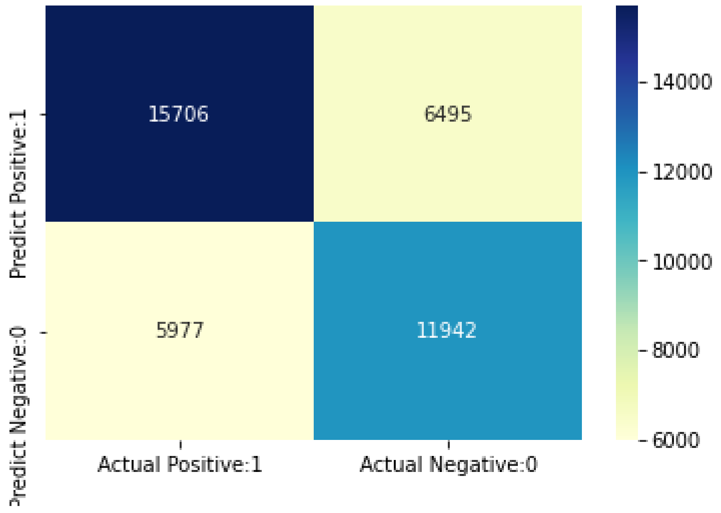
Predicting YouTube Video “Clickbait” Potential and Predicting Metadata Statistics

Pierce Renio

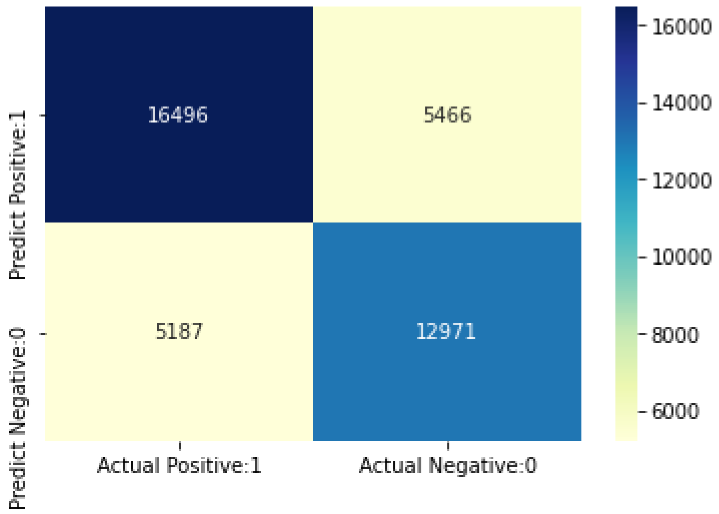
Table of Contents

Abstract/Executive Summary

The modern day allows the people to enjoy many forms of entertainment that were not possible before the turn of the century. However even the more popular methods find it difficult to stay without problems. The popular streaming service, YouTube, is no exception as the problems lie within its user base. The idea of "clickbait" has grown in the last decade especially with the popularity, and almost necessity, of the Internet in everyday households. Clickbait spreads dishonesty sometimes with no way to undo its effects; the values of truth mix with myths to the point where it is now no surprise when Internet users are tricked for the creators benefit. YouTube is one of the biggest victims of this act as the videos on the platform have the possibility of spreading misinformation and/or misleading the viewer. With 500 hours of content uploaded every minute, this problem only gets worse for the platform. While it would be easy to fix this by incorporating an algorithm to detect clickbait, it is impossible to do so at this time. This analysis is to determine if it is possible to change this impossibility by examining the effect the titles videos have on the metadata for the video. The research question "Which words in a title will lead to more views?" will lead us to the answer to this problem. With this analysis, we will be able to determine which words have the highest frequency in titles and determine if they actually lead to greater views overall. With this and the video's metadata, we can then look into the clickbait potential videos have, and create a model that could identify which videos are potentially clickbait. After cleaning the data and ranking the titles based on word frequency, the variable "PotentialClickbait" was used as a classifier for a model to be made. Naive Bayes Classification models were used to determine if a video was clickbait or not.

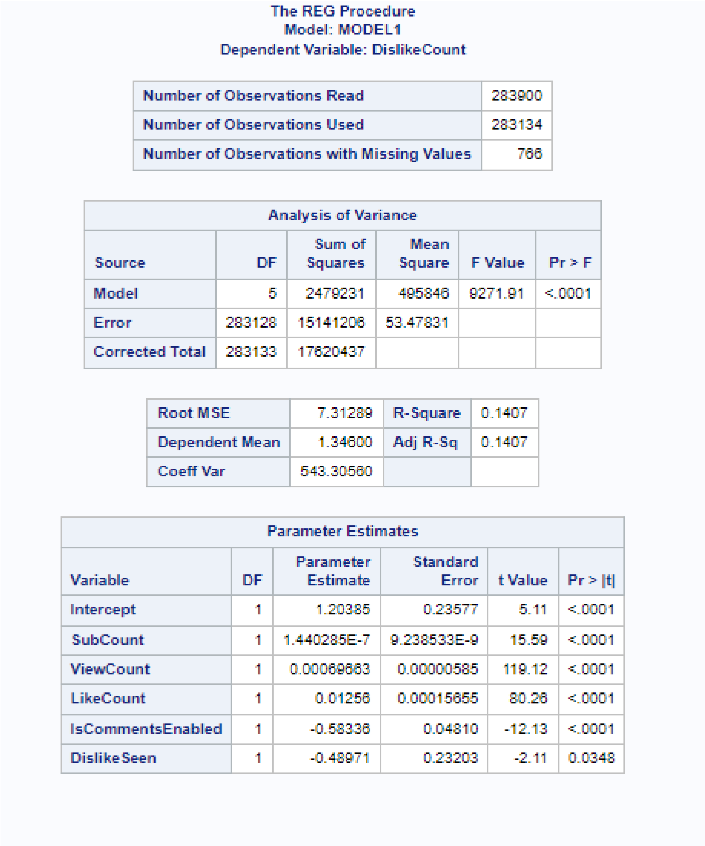


For the first model, only the variables “ViewCount” and “AverageRank” (the average for frequency of words a title has). The accuracy score for this confusion matrix is 0.68913 meaning that the model correctly identified 68.9% of the testing data. The Naive Bayes Classification method was used again but using all of the numerical variables to determine if a better model could be created.



The accuracy score for this new model was 0.73447 meaning that the model correctly identified 73.4% of the testing data. This model was slightly better at predicting if a video was potentially clickbait or not. With these results, it is clear that we are in the right direction for truly identifying a video's potential to be clickbait.

The second idea that became of interest was the removal of the dislike button on the platform. November 2021 marked the removal of the dislike count (the number of dislikes a user could see on a video). The second research question "Does the removal of the dislike count affect the number of views and dislikes a video has?" helps us investigate if the dislike count removal had an effect on people's tendency to interact with a video negatively. Using linear regression and statistical analysis, this question was able to be answered. First a variable was created "DislikeSeen" to identify if a video was published before or after the removal of the dislike count. 2 linear regression models were created, one with and without the new variable. These models were then compared to determine whether or not a viewer's ability to see the dislike count or not actually has an effect on the dislike count.



The REG Procedure
Model: MODEL1
Dependent Variable: DislikeCount

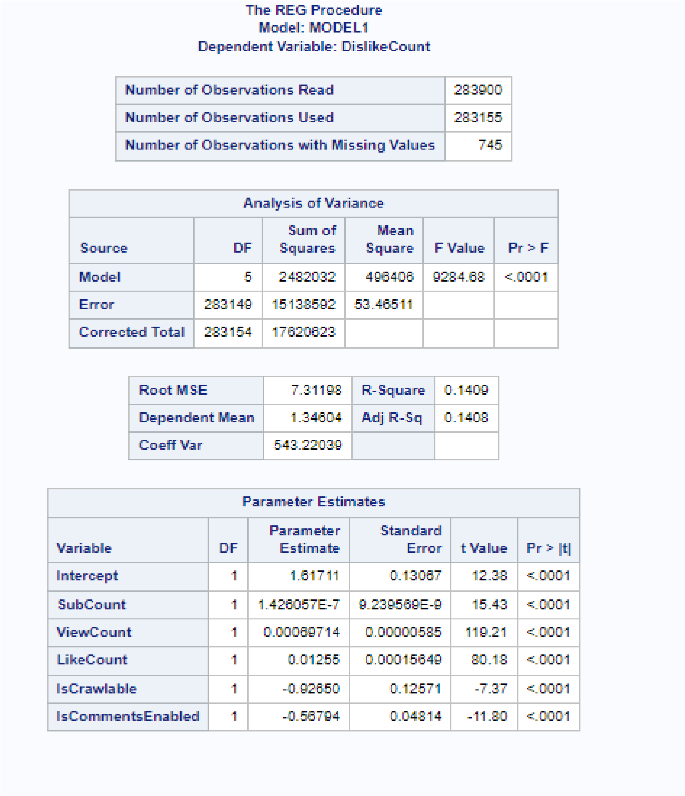
Number of Observations Read	283900
Number of Observations Used	283134
Number of Observations with Missing Values	766

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	2479231	495846	9271.91	<.0001
Error	283128	15141206	53.47831		
Corrected Total	283133	17620437			

Root MSE	7.31289	R-Square	0.1407
Dependent Mean	1.34600	Adj R-Sq	0.1407
Coeff Var	543.30560		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.20385	0.23577	5.11	<.0001
SubCount	1	1.440285E-7	9.238533E-9	15.59	<.0001
ViewCount	1	0.00069663	0.00000585	119.12	<.0001
LikeCount	1	0.01256	0.00015655	80.26	<.0001
IsCommentsEnabled	1	-0.58336	0.04810	-12.13	<.0001
DislikeSeen	1	-0.48971	0.23203	-2.11	0.0348

The model that included the variable “DislikeSeen” had an F-value of 9271.91 with a corresponding p-value of less than 0.0001. The R^2 adjusted value was 0.1407 meaning that 14.07% of the variation in the number of dislikes can be explained by the independent variables.



The REG Procedure
Model: MODEL1
Dependent Variable: DislikeCount

Number of Observations Read	283900
Number of Observations Used	283155
Number of Observations with Missing Values	745

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	2482032	496406	9284.68	<.0001
Error	283149	15138592	53.46511		
Corrected Total	283154	17620623			

Root MSE	7.31198	R-Square	0.1409
Dependent Mean	1.34604	Adj R-Sq	0.1408
Coeff Var	543.22039		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.61711	0.13067	12.38	<.0001
SubCount	1	1.426057E-7	9.239569E-9	15.43	<.0001
ViewCount	1	0.00069714	0.00000585	119.21	<.0001
LikeCount	1	0.01255	0.00015649	80.18	<.0001
IsCrawlable	1	-0.92650	0.12571	-7.37	<.0001
IsCommentsEnabled	1	-0.56794	0.04814	-11.80	<.0001

The model that did not include the variable “DislikeSeen” had an F-value of 9284.68 with a corresponding p-value of less than 0.0001. The R^2 adjusted value was 0.1409 meaning that 14.09% of the variation in the number of dislikes can be explained by the independent variables. With many observations and test statistics that were very close in value, it was tough to determine whether or not the dislike removal had an effect on the dislikes. Statistical testing was then used to further analyze this research question. Using the means for the dislikes before and after the removal of the dislike count, a 2-sample t-test was conducted. The test statistic was -7.0103133 and the p-value was 2.3831905e-12 thus the p-value was smaller than the critical value of 0.001. With this information, we can say that the null hypothesis of the means being equal ($U_{Before} = U_{After}$) can be rejected and we can accept the alternative hypothesis ($U_{Before} \neq$

U_{After}). The average number of dislikes a video had before the removal is not the same as it was after the removal.

This report details each step of the analysis. The model built from research and the information gained will help YouTube have better content as well as be free from misleading and dishonest clickbait.

Project Plan

Project Plan Created by:

Pierce Renio

September 4, 2022

Profile of the Organization

Main Company Details:

Founded - February 14, 2005

Founders - Chad Hurley, Steve Chen, Jawed Karim

Headquarters - San Bruno, California

Parent Company - Alphabet Inc. (GOOGL)

Categories - Internet, Video Hosting Service, Search

Address:

Google LLC, D/B/A YouTube
901 Cherry Ave.
San Bruno, CA 94066
USA

Company Communication:

Fax Number: +1 650-253-0001

Website: www.YouTube.com

Business Description:

YouTube is a type of social media platform where users are able to upload videos and watch videos posted. It is the second most visited website, only second to the Google Search page (www.Google.com). The type of content videos can contain range endlessly, people upload new content daily, up to 500 hours of content per minute as of May 2019. People use Youtube as a source of information: how they get their news, how they get their entertainment, how they get their education, etc. There is no limit to what can be posted, given that the video follows YouTube's video guidelines.

While most accounts on the video streaming platform are individuals, many companies have accounts to which videos are posted as a source of advertising. The producers of the videos typically try and obtain as many views, likes, and comments on their videos as possible. Producing quality content videos results in a bigger audience and fanbase which in return brings more revenue and awareness. Having a high subscriber count is usually desired as having larger subscriber numbers brings in higher numbers for other data such as views. A video is considered, by the Internet's (the collective of people who use the Internet) standard, "good" if the view count is high.

The culture the world has today is due to YouTube's ability to enable the globe to be connected. Trends from the Internet tend to originate from YouTube and normal, everyday people grow into celebrities from their videos posted. People have been able to be more involved with the government and politics as videos posted are available for the public, free of charge.

Financials:

Ticker Symbol - GOOGL

Last Financial Data - February 2021

Revenue - US$28.8 billion (46% increase)

Net Worth - ~US$140-300 billion

No. of Employees - 1001 - 5000 employees (source: LinkedIn)

Key Executives:

CEO/President - Susan Wojcicki

Vice President Engineering - Scott Silver

Vice President Engineering - John Harding

VP of Sales - Tony Nethercutt

Vice President, Marketing - Jodi Ropert

Director of Business Operations CTO - Nathen Crane

Vice President, Operations - Matt Halprin

Director, Technology Solutions - Mike Drake

VP Product Management - Johanna Wright

Major Competitors:

As YouTube increases the amount of differing services, the number of competitors also increases. The organizations that YouTube competes with consist of other video streaming services, social media platforms, movie and television streaming services, and music streaming services. Some of the major competitors YouTube has includes:

Vimeo

Dailymotion

Metacafe

Vevo

Facebook

WhatsApp

Instagram

TikTok

Reddit

Snapchat

Twitter

Tumblr

WeChat

Twitch

Netflix

Hulu

Amazon

Apple

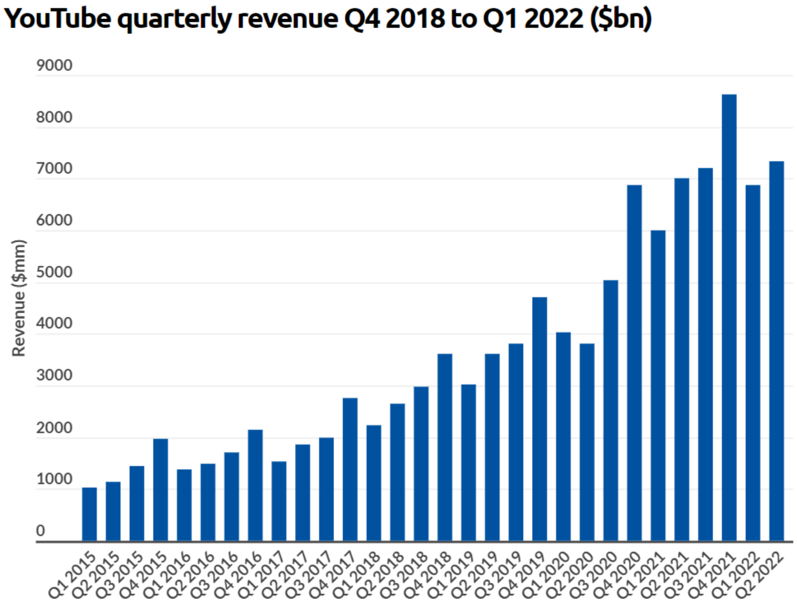
Disney

HBO

Spotify

Business/Analysis Opportunity:

With $28.8 billion revenue being generated in 2021, a 46% increase, and reporting over 30% revenue growth over the last 4 years means the demand for YouTube services are only increasing. This combined with the fact that YouTube's users are only increasing yearly, making it one of the most popular apps in the world, clearly makes YouTube extremely profitable. YouTube has added more services to its platform over time including a premium service for no advertisements, movie renting stream service, and music streaming services.


YouTube quarterly revenue Q4 2018 to Q1 2022 ($bn)

Retrieved from https://www.businessofapps.com/data/youtube-statistics/, September 3rd, 2022

Research Questions

There is much potential for profiting off of YouTube by taking advantage of how their metadata plays into their recommendation system for users. Knowing exactly how YouTube works and what plays into the view, like, and subscriber count could lead to huge success on the platform. The following question will be researched in this project in order to determine what these methods are and if they are as effective as one might expect.

Research Question 1: Which words in a title will lead to more views?

Clickbait titles obviously grant more views on a video due to their eye-catching nature. It is a possible method to increase a video's views and possibly other metadata numbers. With that being stated, are there certain clickbait words that are more effective at grabbing people's attention more than others? Certain words might stand out amongst others in the title of a video, and by looking at the titles of videos we can determine if certain words are more effective than others at bringing in more views.

Research Question 2: Does the removal of the dislike count affect the number of views and dislikes a video has?

Starting November 2021, YouTube removed its dislike button count for viewers to see. People are still allowed to dislike a video but they can no longer see how many dislikes a video has. With users now not able to view the dislike count, will this influence the average dislike amount on videos? Additionally, does this removal change other metadata numbers such as the like count and view count?

Research Question 3: Do clickbait video titles have more of an influence on the like count or dislike count?

Over time, the idea of "clickbait titles" has grown more popular on the Internet, including the video streaming platform. The purpose of these excessive, often misleading, descriptions is to gain more views on a video. Are people aware of this idea, and if so do they actually like this type of content? We can compare the ratio of likes and dislikes to certain clickbait titles to see if these clickbait videos are actually doing their job of getting people to click on the video.

<u>Hypothesis:</u>

Hypothesis 1: Certain "Clickbait words" will lead to a higher view count.

By analyzing the text of a YouTube video title, there will exist particular keywords that lead to greater views. Identifying these keywords will allow us to know which words can be used to take advantage of YouTube's system.

Hypothesis 2: The removal of the dislike count has led to a decrease in dislikes.

People like to follow what they see, especially if it is anonymously, behind a screen, where no one can judge their actions thus making their action more authentic. Not seeing the dislike count will let the dislikes from users be more authentic rather than it being just because they see a high dislike count. This hypothesis is intended to be tested.

Hypothesis 3: Clickbait video titles will have a greater influence on the dislike count than the like count.

By this modern age of technology, most seasoned Internet users are familiar with clickbait content. They recognize it, whether it be slight or on an obvious grandiose (almost embarrassing) scale. Being aware of the clickbait content is the first step to knowing that it is unwanted and thus disliked.

Data

The data consists of over 4 billion observations of videos on YouTube. Each observation provides metadata of the video including key variables such as view count, dislike count, like count, number of subscribers, date of upload, and video title. The size of the data is massive and will not be fully used, however this works perfectly as the data is provided to the user through multiple parts. Being that the data is randomized, it makes it easy to section the data into testing and training data.

View Count

This is arguably the one of the most important parts of the data, as well as one of the most important numbers that matters to a "YouTuber", one who uploads on YouTube regularly. This number influences revenue and awareness more than any other data so it is the statistic we will measure when it comes to certain words in a video title.

Like Count/Dislike Count

This determines if a user "likes" or "dislikes" a video. This action will provide "better" videos for the user's recommendation page as it will suggest videos either closely related to the videos they liked (possibly videos from the same person who uploaded the original liked video) or videos that branch out in other directions regarding content.

Number of Subscribers

The other arguably most important number for YouTubers. Essentially, the concept for a subscriber is to support those of which they are subscribed to, including providing their view for a video uploaded. Therefore 1,000 subscribers would guarantee at least 1,000 views for each video uploaded after that YouTuber reaches 1,000 subscribers. While not as important as view count for immediate results regarding statistical analysis, subscriber count is useful when looking at long term effects for both the user's and the YouTubers.

Date of Upload

Usually not too important for certain measurements, however it can be used to explain why certain videos become popular. In our study, we will use this to determine if the video was uploaded before or after the removal of the dislike count. Being able to separate the videos into before and after the removal of the button, we are easily able to analyze this part of the data.

Video Title

This portion of the data is extremely important for the YouTuber to create as it is one of the first things users see before clicking on the video to watch it. The words and text in the title will directly correlate with the amount of views a video will have.

Measurements

Certain, and proper, measurements will be collected for each hypothesis tested. In order to look into clickbait titles, we will need titles for videos as well as the number of views, likes, and dislikes. To see if the number of dislikes have changed with the removal of the dislike count, we will obviously need the amount of dislikes a video has and the date the video was uploaded to determine if it was uploaded before or after the removal of the dislike count. Fortunately, the data found for this project included all of these aspects that are needed in order to properly test the hypotheses.

Methodology

For research question 1, we are trying to determine which words are more effective at bringing in more views for a video. In order to test this, we will use both structured and unstructured data as we will need both the view count of a video (structured) and the title of a video (which has no format thus being unstructured).

For research question 2, we are trying to determine if the removal of the dislike count has an effect on the number of dislikes. For this we will want more structured data as we only need the dislike count as well as the date of which the video was uploaded. Since the date of the upload has a format to it, it will be structured.

For research question 3, we are attempting to discover if clickbait titles have an effect on the amount of likes and dislikes a video has. This question requires both structured and unstructured data, like research question 1, as we need both the like and dislike count, as well as the titles of the video which are unstructured.

The structured data will be easy to work with as they are mostly positive integers, with exception to the dates of which the video was uploaded. Easy numbers to work with here allows us to analyze the variables with no difficulty as we can plot and statistically work with the numbers with no problem.

The unstructured data will have a different approach than the structured data as the format for the titles of the videos does not exist. The Naive Bayes Classifier is the model that comes to mind as we can determine if a video is “clickbait” based on the words in the title. However, other models not listed are not necessarily excluded in the modeling process.

Computational Methods and Outputs

With classification, we believe that the Naive Bayes Classification will be the most effective model for research questions 1 and 3 as they deal with the titles of the videos which are unstructured data. Due to the fact that the data is segmented into many parts, it makes it helpful as the models we create can be used for our 80/20 data split as well as using it on the other segmented data parts to test how effective the model is.

For the structured data in research question 2, regression techniques would be the first part of the analysis as comparing the variables this way would be most effective in detecting a relationship between the variables if any at all. Being able to see if there was a change in the dislikes for videos before the dislike count was removed to after the count was removed can be done with basic statistical analysis.

Output Summaries

Research Question 1: Which words in a title will lead to more views?

The analysis will be able to identify which words will lead to more views for a video on YouTube. A sentiment analysis would be used to identify this. We should be able to have a table that reveals the words that are most used in video titles and which words are most effective and exactly how effective they are in comparison to the other words that are effective. For visual purposes, we can also provide a word cloud that effectively displays which words are more effective at getting views.

Research Question 2: Does the removal of the dislike count affect the number of views and dislikes a video has?

For this analysis, regression charts can be provided to see the relationship between the number of views and dislikes a video has before and after the dislike count was removed. By analyzing the ratio before and after the change, it will be determined whether or not the removal of the count has a statistically significant effect on the views and dislikes a video has.

Research Question 3: Do clickbait video titles have more of an influence on the like count or dislike count?

The sentiment analysis used in research question 1 can be used here to determine if the like count or dislike count is influenced more by the video titles. We can use the top used words in video titles and most effective words used to see if they have an effect on the like and dislike count. If the word is successful in gaining views, this does not mean they are useful in gaining likes. Being that they might lead to dislikes, we can see if the number of views is worth the number of dislikes or likes a video has. With this in mind, we can use a table that compares the words that are both most used and most effective in titles to the number of likes and dislikes the video ends up having.

Campaign Implementation

YouTube has been a growing organization and will continue to be one of the biggest video streaming platforms, amongst other provided services. With an increasing number of users every year, more videos will be published and even more videos will be streamed on the platform. It is important to understand how YouTube works as both a consuming user and as a producing content creator; users should know that the content they watch will not waste their time with clickbait and producers should know what they should name their videos if they wish to make ethically genuine content.

Each of the research questions allows the platform to benefit by stopping the production of bad content and increasing the production of quality content that both users and producers enjoy. Being able to develop a model for research question 1, “Which words in a title will lead to more views?” allows users and content creators to avoid titles that manipulate the users into clicking on videos for a cheap view, which ultimately leads to more people watching the video.

Knowing if the dislike count was actually an effective tool of users picking out bad content or not can be answered with research question 2, “Does the removal of the dislike count affect the number of views and dislikes a video has?”. The purpose of the dislike count is for users to actively voice their opinion on a video they genuinely dislike, rather than following the masses and possibly disliking quality content simply because the dislike count is high.

Developing a prediction model for clickbait titles and the like/dislike count will allow us to know if users can differentiate between what is clickbait and what is authentic content, and if they dislike clickbait content. Research question 2, “Do clickbait video titles have more of an influence on the like count or dislike count?” will allow us to know if YouTube’s user base is aware of the efficacy of their actions. Being able to stay united against clickbait is one of the main ways the community can deal with bad content and work towards making more authentic videos.

Review of the Literature

YouTube, the social media and video streaming platform, has gained enormous popularity since its launching in 2005 as it is one of the world's most popular digital media platforms. The user base, both video watchers and content creators, have grown tremendously as there seems to be a video for just about anyone; the range of content on YouTube is never limited as new content is discussed and made every day. Content creators not only use new content as a way to promote their channel but also use techniques that involve manipulating YouTube's algorithm with metadata. Exploring statistical techniques used for YouTube statistics is valuable as it will provide a basis for further exploration on YouTube metadata. Additionally exploring YouTube's algorithm and metadata, as well as YouTube culture, is valuable as it may provide insight for what both content creators and video watchers are thinking about and possible explanations for their actions.

Lau Tian Rui, Faculty of the Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, explains how research for YouTube data has been "receiving growing attention such as…sentiment analysis". Many models were used to predict view count for YouTube videos as view counts are imperative for modeling and attempting to characterize video watchers. Sentiment analysis' are used with the comments of a video to determine views; using the same statistical techniques with video titles will also prove to be fruitful.

Video views are meant to be based on the content of the video but those who have been using YouTube's platform know by now that any video could be misleading based on the title. Peter Braun, from the University of Manitoba, Winnipeg, elaborates "The users were misled into clicking the video by the content creator" . Braun continues by detailing the consequences of this “damage”, “the content creator earns a small sum of earnings, and (most importantly) it shows up in users’ history” which then offsets the algorithms of the recommendation systems. This effect spirals as the video appears in other people’s recommendations and the cycle of people being tricked and the video becoming “popular” occurs again. It is important to viewers that clickbait videos are gone as their time might have been wasted however having a solution to this is not easy. Braun details how users could remove the videos from their watch history but doing this is not practical as it cannot be automated as well as the fact that the video still remains on recommendation lists for viewers. Being able to create a statistical model that could predict clickbait videos would be in the best interest of the viewers as a model would remove videos from their recommendation list before it is clicked on, watched, and influential on the rest of the recommendations.

Celebrity status was given to those who are famously well known throughout the world, being so popular that they are topics of conversation by people who do not even know them personally. Jane Arthurs, from Middlesex University, explains that the idea of building your own brand through self-presentation and becoming a “micro-celebrity” has trended on YouTube so much that the idea of anyone doing it is actually a plausible idea. The term “vlogging”, a shortened term for video logging, also grew in popularity; vlogging is the act of recording one’s daily life and posting it. These videos became part of the YouTube culture in the last decade as content creators try to promote their brand as much as possible despite the decrease in video quality. Arthurs explains how the strategies used by “successful beauty vloggers…are influenced by their knowledge and assumptions about the workings of the algorithm”. Content creators know that in order to become successful on the video streaming platform, they must do more than just make quality content.

Exploratory Data Analysis

Introduction

For this project, the data was found on the subreddit: r/datasets where there exists an abundance of data sets. This data was collected by the Archive Team from YouTube in November/December 2021 and the data set contains over 4.5 billion observations. Since the data is massive, there are a number of downloads all containing different parts of the data in the order that the video was fetched from the site. Other than the order of the fetch date, all of the other parts of the data are completely random which makes this convenient to work with. The file that I downloaded, which was the first listed link, was 50.1 MB and contained 335,591 observations with 16 variables (the Data Cleaning section will go more into depth about the observations).

VideoID: Identification of the video. This identity can be found in the omnibox for a YouTube video after the “watch?v=” string. This part of the data is not needed for the analysis that will be conducted.

UploadDate: The date at which the video was uploaded to YouTube. The values are in the format YYYYMMDD (year, month, day). There are some live videos in the data so some observations where the video is actually a live video will be “erroneous”.

FetchedDate: The date at which the video was webscrapped from YouTube. The values are in the format YYYYMMDDHH24MMSS (year, month, day, military hour, minute, second). The precision in this value is not needed for the analysis so it will need to be cut down to match the format that the variable UploadDate follows.

UploaderID: Each YouTube channel has a specific identification that is associated with their account. This value is much like the VideoID and is not needed for this analysis.

SubCount: The number of subscribers the uploader of the video has at the time the video was fetched. A value of -1 indicates that the number of subscribers is hidden from public view. These observations will not be used as they will not be helpful.

ViewCount: The number of views a video contains at the time it was fetched.

LikeCount: The number of likes a video contains at the time it was fetched.

DislikeCount: The number of dislikes a video contains at the time it was fetched.

IsCrawlable: A video is able to be made public or private. A video can be accessed by anyone if it is public and only accessed by certain people if it is private. If a video is private/unlisted, it is not crawlable meaning you cannot search on the website and find the video. (1 for yes, 0 for unlisted)

IsAgeLimit: Binary value if this video is age restricted. (1 for yes, 0 for no)

IsLiveContent: Binary value if video is live content rather than uploaded. (1 for yes, 0 for no)

HasSubtitles: Binary value if subtitles are enabled for this video. (1 for yes, 0 for no)

IsCommentsEnabled: Binary value if comments are enabled for this video. (1 for yes, 0 for no)

IsAdsEnabled: Binary value if ads are enabled for this video. (1 for yes, 0 for no)

Title: The title of the video.

Uploader: The name of the YouTuber who posted the video.

I made 3 new variables that may be useful for the analysis: “year”, “month”, and “day” which all pertain to the “UploadDate” variable. These new variables are just the “UploadDate” variable split into the 3 components the format is in.

Data Cleaning

I chose to use a Jupyter Notebook to clean the data as using python for both data and string manipulation is efficient. Using python and the pandas (Python Data Analysis Library) package, the data was able to be cleaned. First I took a look at the data as a whole to grasp what the data as a whole looked like. Figure 1 shows the table for the first few rows of the data. There are the 16 variables that were described previously and 335,591 observations. Expecting each and every observation of this data frame to be perfect is nonsensical so we looked into removing some of these unhelpful observations. Figure 2 displays the removal of observations that contain “n/a” in any column. There were not many, which is surprising of a data set this large, but removing these values was just the beginning. The VideoID variable was removed entirely. Then the variable UploadDate was cleaned so that only numeric values existed. From this point, the three variables, year, month, and day, were created. Then the variable FetchedDate was changed so that the format for the values matched the format for the UploadDate values. To complete this particular process, another sweep of “n/a” values was conducted as there were some present. Likewise with the “n/a” values and the non-numerical values found in UploadDate, there might have been the same problem found in other variables. These variables were also swept. Lastly, the SubCount variable needed to have its values of -1 removed.

Using the value_counts method, we are able to get a sneak peak of some of the variables before we take a much closer in-depth look at the data. This can be found in Figure 3. For the variable IsCrawlable, 98.9% of the videos in the cleaned data are not unlisted. It might be wise to look at the crawlable videos separately but not without also looking at the unlisted videos as well. For the variable IsAgeLimit, an even higher 99.58% of the videos are not age restricted. This makes sense as if a video is age restricted then the video cannot reach far out to other people thus not gaining as many views and/or likes. I would like to explore this further later. For the variable IsLiveContent, 93.6 % of the videos are not live, which is surprising to me as I expected there to be less live content. The idea of live content works very differently to uploaded content so it might be worthwhile to look into the differences between the two types of videos and their corresponding metadata. For the variable HasSubtitles, only 41.56% of the videos actually have subtitles. The idea of having subtitles on videos seems controversial, in a light way, but not everyone agrees with how a video should be watched regarding the text that appears on the screen. For the video IsAdsEnabled, only 10.38% of the videos have ads on them. This is surprising because ads are a way to make income on the video platform, this begs the question why aren't more people putting ads on their videos? This could be possibly due to the fact that one needs to reach a certain "level" on YouTube as content creator before one starts to receive benefits such as income. It will be interesting to see how this variable correlates with both SubCount and ViewCount.

Finally, the data has been cleaned. The total number of variables increased to 18 and the number of observations decreased to 298,850, which is 89.05% of the original number of observations. The updated data frame was then downloaded into a csv file so that the data analysis could be continued.

Data Analysis

The program that was used for the data analysis is SAS (Statistical Analysis System) as it is effective at looking at the data as a whole, the variables individually, as well as looking at the possible correlations/relationships between variables.

Using a proc means and a proc univariate statement in Figure 4, we are able to see the min, max, mean, standard deviation, and histogram for each numerical variable. The mean and standard deviation for the variables revolving around the date won't make sense since the variables are in a specific format and not actually a number used for calculation. However it is important to note the minimum and maximum, the earliest video in the data set is November 2nd, 2005 and the most recent video in the data set is January 19th, 2022. This data set contains videos over a span of the last 17 years and this should be important because the number of views and subscribers a video/YouTuber has will definitely increase as time goes on because the number of users YouTube had back in 2005 is significantly lower than how many there are over 1.5 decades later.

Additionally, having videos the dates to at least November 2021 is important because that is the year and month that YouTube decided to make the dislike button hidden from public view. We are able to look more into the relationship between videos before and after the dislike count was removed and the dislike count of a video. The average SubCount, and ViewCount of a video have great differences which is interesting. Average SubCount is 129580 and average view count is 16467. A subscriber for a YouTube channel basically means you are "guaranteed" 1 view for a video uploaded, so one would think that 129580 subscribers would translate to 129580 views on a video. Of course this is just the averages so it would be beneficial to look into the relationship between subscribers and view counts for a video to see if this pattern also follows the one we see with the means.

The distribution for the other variables are not normally distributed as they are mostly skewed heavily right. Distribution charts for SubCount, ViewCount, LikeCount, and DislikeCount (Figures 15-18) demonstrate this pattern. The distribution for IsCrawlable, IsAgeLimit, IsLiveContent, isCommentsEnabled, and IsAdsEnabled are either distributed largely in one of the binary values or the other (Figures 19-21,23-24), only the variable HasSubtitles (Figure 22) is distributed fairly evenly (about 60/40).

We first created a new variable in SAS called "dis_per_view" which is the dislikes per view on a video. Next, we separated the data into 2 parts: before the removal of the dislike count (October 2021 or earlier) and after the removal (November 2021 or after).

Using the proc sgplot command in SAS we can plot DislikeCount and ViewCount for both sets of data. This can be seen in both Figures 5 and 6. The number of observations for the

YouTubeBefore data is greater than the number of observations found in the YouTubeAfter data, and this can be seen when comparing the two scatter plots. Both plots appear to be clustered near the origin, this is understandable as most of the videos have less views and even less dislikes. However it appears that the YouTubeBefore data has more points that have higher ViewCount and lower DislikeCount. We will check the correlation coefficient to further examine this comparison. This can be found in Figures 7 and 8. The correlation found in the YouTubeAfter data is 0.89701, which is greater than the correlation of 0.64447 found for the YouTubeBefore data. It is worth noting that both of the p-values for the correlations were less than 0.0001. However it is also worth noting that despite the correlation coefficient being higher for the YouTubeAfter data, the number of observations for the YouTubeAfter data is only 1096 while the number of observations for the YouTubeBefore data is 296984 which is over 270 times the first number of observations. A 2 sample t-test would be extremely useful for later to determine if the two population means are equal, or if one is actually statistically greater than the other.

We now look at the potential relationship between the number of views a video has and the number of subscribers the YouTuber who uploaded the video has. Figure 9 displays the scatterplot between ViewCount and SubCount. The number of subscribers should equal the number of views a video has but the scatter plot does not show a linear relationship at all. If anything, it seems almost negative. Additionally the points are almost divided where some of them follow the relationship that ViewCount and SubCount are expected to have, and some of them follow the opposite where YouTubers with a higher SubCount get hardly any views relative to their SubCount. We then look into the correlation coefficient between the two. The value for the correlation coefficient is 0.06457 which does not give much of a linear relationship at all but does indicate that on average the points tend to be positive. The p-value is less than 0.0001 thus it is a reliable source.

It is also worth looking into the variable IsCrawlable for these two variables. If we group only the public videos together, do they follow the expected trend between SubCount and ViewCount more so than if it is grouped with the unlisted videos? We first divide the data into 2 parts: if the video is crawlable or not (where IsCrawlable is set to 1 and another where it is set to 0). The data was also focused on the views and subscribers being less than or equal to 1000 as the data is spread very widely but concentrated heavily on smaller values.

Using the same proc sgplot, we can now look into the relationship between SubCount and ViewCount on a more public and private scale. Figures 11 and 12 display the scatterplots. The scatter plot for YouTubeCrawl paints the graph all around but less so when SubCount and ViewCount increase. The scatter plot for YouTubeUnlisted shows a different pattern as there are not nearly as many observations; there are more observations near the origin. We can get a much better idea of the issue at hand if we look at the correlation coefficients. Figures 13 and 14 detail the procedures. The coefficient for the crawl group is 0.11526 and the coefficient for the unlisted group is 0.21447. It seems like the answer to this question is that removing the unlisted videos from the group does have an effect on the expected trend but only slightly. This makes sense because even though unlisted videos have less views than public videos do, there are not nearly as many unlisted videos in the data set (n = 3569) so that even the removal of these videos only has a slight effect on the overall relationship between SubCount and ViewCount.

The information gained in this analysis will be extremely helpful to the understanding of the research questions after conducting the statistical analysis on the data later. While the future analysis will be the main focus of answering the research questions, the research done here sheds light upon the rest of the data not being used in the statistical analysis as the sentiment analysis will further explain ViewCount phenomenon (the data used there is not able to be used in a numerical sense here), as well as looking into several of the possible relationships that will be further explored on a statistical level (i.e DislikeCount and ViewCount on UploadDate). The models that will be built will have the foundation to explain why each method and variable was chosen.

Methods

RQ1 - Research Question 1: Which words in a title will lead to more views?

Sentiment Analysis and Naive Bayes Classification

Each YouTube video has a title as a part of its metadata and while numerical metadata (view counts, likes and dislikes, etc) has plenty of methods that can be used, unstructured data such as these titles makes it difficult to do the same analysis on it. Being able to do a sentiment analysis on the YouTube titles allowed us to possibly categorize certain words used in videos, such as which words are more effective at bringing attention to videos and thus increase the views.

Through our EDA, it was discovered that many video titles are not in English. As the videos are in many different languages, it is important to perform this Sentiment Analysis on videos that the algorithm can understand, thus the video titles in English were used for this sentiment analysis. An R package called "textcat" was useful in detecting which language the YouTube video is in as it can detect 74 languages. The Journal of Statistical Software explains, "Identifying the language used will typically be the first step in most natural language processing tasks" thus this package was helpful in understanding which titles are in English. Another variable in the data was created, Language, and another data frame was created from the original that includes only the videos with "English" as the value for the Language variable.

Unlike typical Naive Bayes Classification, this data set does not include if the video is actually "clickbait" or not. It would have been nice to assign the variable ourselves through some other algorithm so that the Naive Bayes Classification can work with it, but that would defeat the purpose of the Naive Bayes Classification. The idea to get around this was to first identify the most popular words used in the YouTube video titles. This list of words does not include "throw away" words such as the words "and", "or", "the", etc. They are so common in videos that they do not help determine if a video gains more or less views. Additionally, we had to stem words so they would not be considered two individual words. The words were given a rank based on how common they are used, 1 being the most common, 2 being the 2nd most common, etc.

Then the average rank was found for each YouTube title; all the word ranks summed up and divided by the number of words in the title. Here we created another variable: AverageRank. Looking at the summary statistics, a baseline can be established for clickbait. If the AverageRank for a video is 1 standard deviation above the mean and the title contains at least one of the top 50 most used words, it was assigned a 1 (for yes) to the new variable, PotentialClickbait. The Naive Bayes Classification algorithm was used to determine if a video was potentially clickbait based on the amount of views the video has, as well as the average ranking of the video title. The new variable allowed us to use the Naive Bayes Classification to fit the model with the training data, then apply the model to the test data. A confusion matrix and accuracy score were given to both the training set and the test set to determine the utility of the model.

RQ2 - Research Question 2: Does the removal of the dislike count affect the number of views and dislikes a video has?

Preparation

Some of the preparation for this research question, mostly the data cleaning, was done through the EDA as the data was prepped in a way where the dates had a consistent format with each other. Additionally, the year, month, and day variables help separate the format that the variables “UploadDate” and “FetchedDate” are in. The EDA discovered that the correlation for DislikeCount and ViewCount for the YouTubeAfter filtered data (YouTube videos where the UploadDate was after the removal of the dislike count) was greater than the correlation between DislikeCount and ViewCount for the YouTubeBefore filtered data (YouTube videos where the UploadDate was before the removal of the dislike count). Although the sample size for YouTubeBefore is 270 times the sample size of YouTubeAfter, other methods might be able to provide insight. We can compare the two filtered data frames as well as seeing how DislikeCount and ViewCount do on the data as a whole without filtering.

Modeling Techniques:

- Linear Regression
  - Creates a least squares regression line model that determines the predicted y value (dislikes) based on the independent variable (views)
  - Can do this for each data frame to see if dislikes are more likely to be influenced by views before and after the dislike count removal.
  - Tested the utility of the model using the F-test, R^2 value, as well as the individual coefficient for the parameter (B1).
- Logistic Regression

    - Similarly to Linear Regression, creates a logit model for predictive analysis, we can see if altering the independent variable affects log-odds for the video being uploaded before or after the removal of the dislike count.
    - Tested the utility of the model using the F-test, R^2 value, as well as the individual coefficient for the parameter (B1).
- Naive Bayes Classification
    - A text classification staple; in order to make predictions, the Naive Bayes Classification model applies Bayes Theorem on the independent variables to calculate probabilities that are paired with each variable. The model considers all independent variables to be unrelated to other independent variables (thus being "Naive")
    - Tested the utility of the model by using a confusion matrix and an accuracy score.
- K-Nearest Neighbors
    - Allows predictions between whether a video was uploaded before or after the removal of the dislike count by looking at the closest points in the data set.
    - Tested the utility of the model using inRMSE for the training data and outRMSE for the test data. Used the smallest outRMSE to optimize k.

Other Techniques:

- Statistical Analysis (2-sample t-test)

- ○ We have 2 populations (before and after the removal of the dislike count) thus we can see if the population means for the dislikes and views are equal or not.

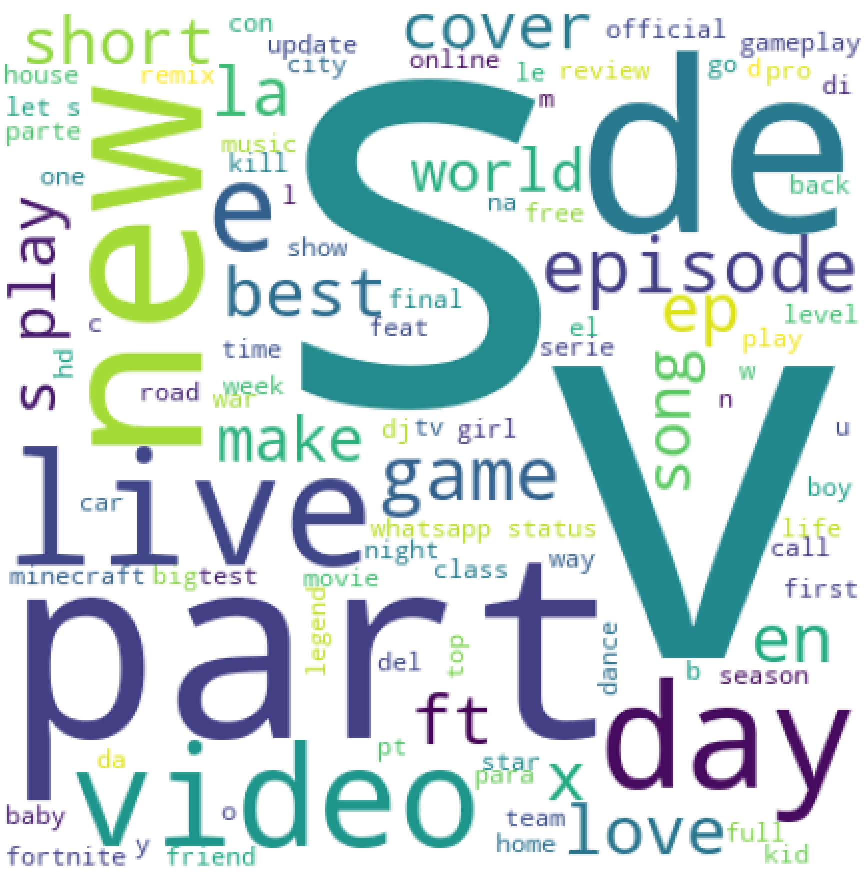
Data Visualizations/Analysis

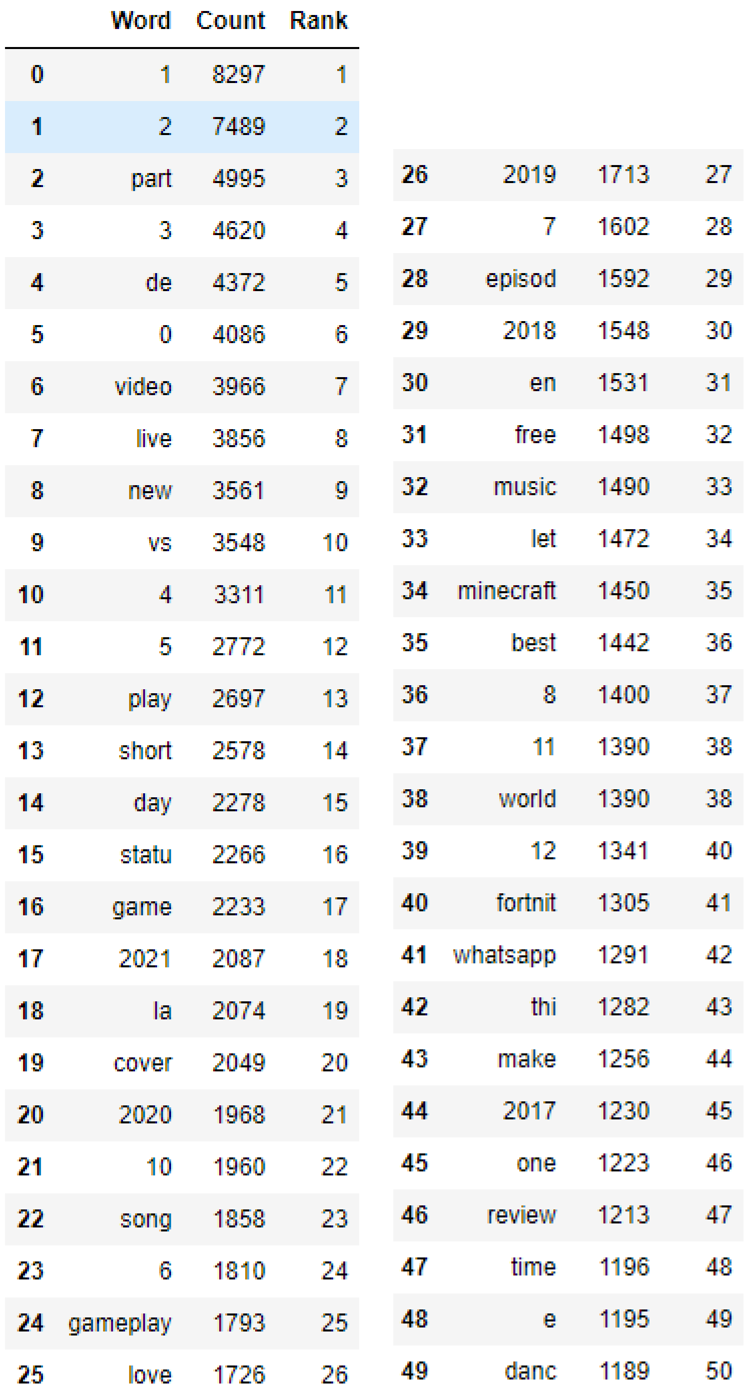
Research question 3 “Do clickbait video titles have more of an influence on the like count or dislike count?” was not pursued further for research but could be a question for further investigation.

Research Question 1 - Which words in a title will lead to more views?

Sentiment Analysis/Naive Bayes

The first task for the analysis was to run a sentiment analysis on the titles of the YouTube videos using the Naive Bayes Classification method. Knowing if a video is clickbait or not is impossible without actually watching the video, and the idea of what is considered clickbait is subjective, so to be able to perform the Naive Bayes method, we needed to create a variable named “PotentialClickbait” in order to determine if a video has the potential to be clickbait based on other figures. This was done by separating out the non-English videos from the data as performing the tokenization on video titles would not have worked as well if this was not completed since videos were in many different languages. After this, we were able to see which words were the most used in video titles. There were many “stop words” in the data which did not help the title’s performance so they were removed before any more analysis was done. The table and word cloud displays the most common words


short
con
update
city
cover
official
gameplay
house
remix
la
online
le
review
go
d
pro
di
m
let s
parte
new
s
de
music
kill
world
back
one
l
na
free
e
play
show
best
episode
final
feat
ep
level
c
el
time
serie
play
hd
s
w
road
week
war
song
n
make
dj
tv
girl
u
game
live
car
boy
whatsapp status
life
night
call
way
minecraft
big
test
movie
class
first
v
legend
top
dance
en
part
del
b
season
ft
day
da
pt
star
para
x
video
love
baby
o
team
full
y
home
fortnite
friend
kid

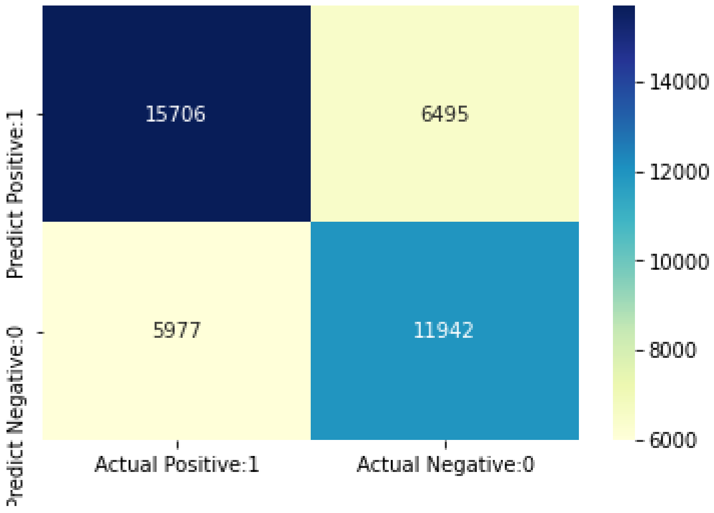


	Word	Count	Rank
0	1	8297	1
1	2	7489	2
2	part	4995	3
3	3	4620	4
4	de	4372	5
5	0	4086	6
6	video	3966	7
7	live	3856	8
8	new	3561	9
9	vs	3548	10
10	4	3311	11
11	5	2772	12
12	play	2697	13
13	short	2578	14
14	day	2278	15
15	statu	2266	16
16	game	2233	17
17	2021	2087	18
18	la	2074	19
19	cover	2049	20
20	2020	1968	21
21	10	1960	22
22	song	1858	23
23	6	1810	24
24	gameplay	1793	25
25	love	1726	26
26	2019	1713	27
27	7	1602	28
28	episod	1592	29
29	2018	1548	30
30	en	1531	31
31	free	1498	32
32	music	1490	33
33	let	1472	34
34	minecraft	1450	35
35	best	1442	36
36	8	1400	37
37	11	1390	38
38	world	1390	38
39	12	1341	40
40	fortnit	1305	41
41	whatsapp	1291	42
42	thi	1282	43
43	make	1256	44
44	2017	1230	45
45	one	1223	46
46	review	1213	47
47	time	1196	48
48	e	1195	49
49	danc	1189	50

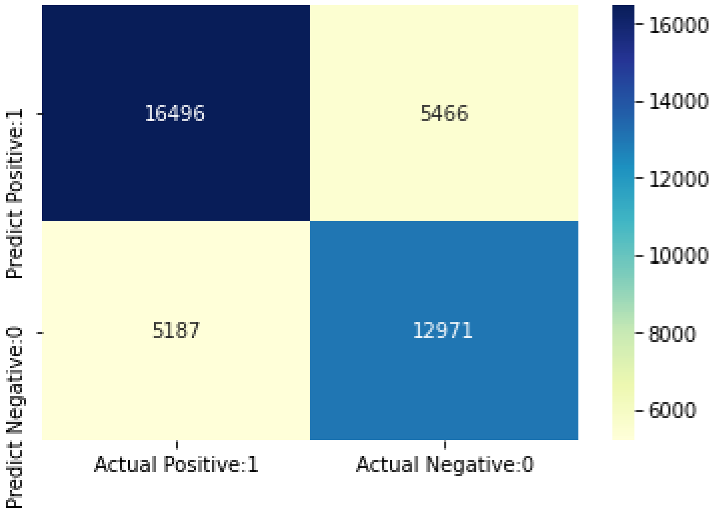
The integers in the table are not displayed in the word cloud which is why the word cloud and the table contain slightly different results. It is important to consider the integers in the titles as these integers could be detailing parts of the title which are contextually important so they were not removed.

With the most frequent words at our disposal, we are able to create a variable titled “Rank” which will be used to calculate the variable “AverageRank” which takes each word in the tokenized titles and averages the ranks of all the words. This value will help determine the value for the new target variable “PotentialClickbait”. The original idea of using the standard deviation as a metric for “ViewCount” and “AverageRank” to determine “PotentialRank” was changed to the median because the standard deviation was greater than the mean for these variables because the data has abnormal distribution. “PotentialClickbait” was set to a value of 1 if the video’s “ViewCount” was greater than the median and the “AverageRank” was less than the median or if the “ViewCount” was greater than the “SubCount”.

With “PotentialClickbait” being set, the Naive Bayes Classification method can be performed. Using the variables “ViewCount” and “AverageRank” as the predictors, a model was completed for determining “PotentialClickbait”. After fitting the model on the training data, the model was used on the training data to see its performance before using the testing data. The accuracy score for the training set was 0.86525. For the testing set, the model results can be easily viewed with a heatmap that displays the confusion matrix.



The accuracy score for this confusion matrix is 0.68913 which is much smaller than the confusion matrix for the training data. This could be due to the fact that the model was overfitting/underfitting the data. The Naive Bayes Classification method was used again but using many more variables to determine if a better model could be created.



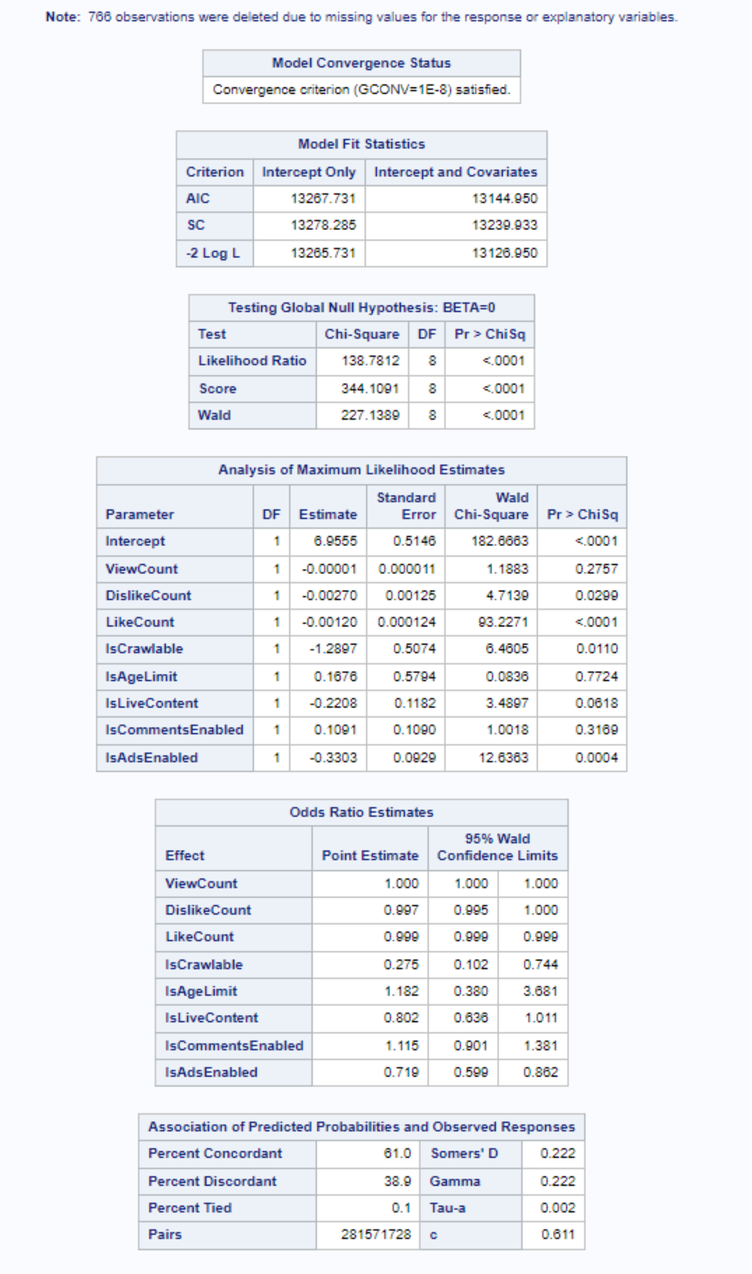
The accuracy score for this new model was 0.73447. The amount of true positives and true negatives have been increased in this new model which is an indicator that the original model was underfitting the data. If the original data set contained more variables, better models could possibly be created.

Research Question 2: Does the removal of the dislike count affect the number of views and dislikes a video has?

Linear Regression/Logistic Regression

The dislike count removal raises questions regarding the number of dislikes and views a video receives. We want to find out if there is a difference for these statistics before and after the removal of the dislike count. For this, the variable “DislikeSeen” was created, a value of 1 is given if the video was published on the streaming source before the dislike count were removed and a 0 if the count cannot be seen (November 2021 and forwards receives a 0)

First we will use Logistic regression to determine if DislikeSeen can be predicted. The odds ratio estimates are in the following table and the ROC plot is also displayed:



Note: 766 observations were deleted due to missing values for the response or explanatory variables.

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	13267.731	13144.950
SC	13278.285	13239.933
-2 Log L	13265.731	13126.950

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	138.7812	8	<.0001
Score	344.1091	8	<.0001
Wald	227.1389	8	<.0001

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	6.9555	0.5146	182.6663	<.0001
ViewCount	1	-0.00001	0.000011	1.1883	0.2757
DislikeCount	1	-0.00270	0.00125	4.7139	0.0299
LikeCount	1	-0.00120	0.000124	93.2271	<.0001
IsCrawlable	1	-1.2897	0.5074	6.4605	0.0110
IsAgeLimit	1	0.1676	0.5794	0.0836	0.7724
IsLiveContent	1	-0.2208	0.1182	3.4897	0.0618
IsCommentsEnabled	1	0.1091	0.1090	1.0018	0.3169
IsAdsEnabled	1	-0.3303	0.0929	12.6363	0.0004

Odds Ratio Estimates

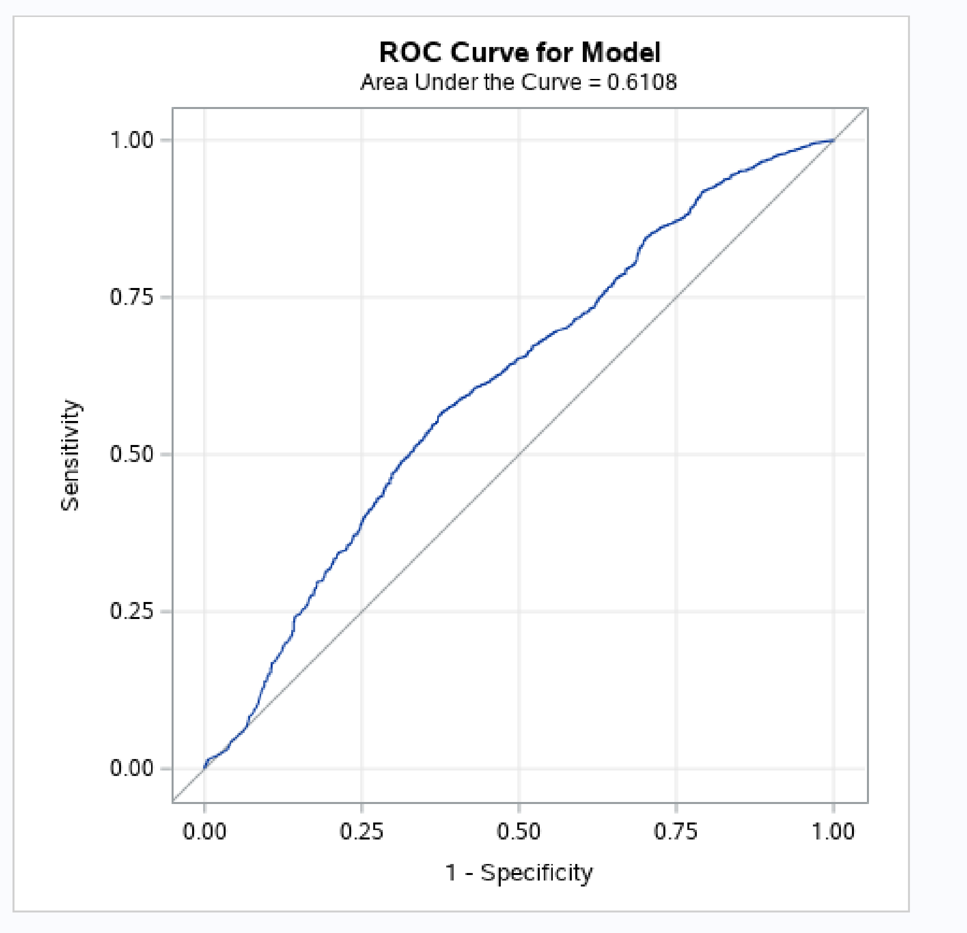
Effect	Point Estimate	95% Wald Confidence Limits	
ViewCount	1.000	1.000	1.000
DislikeCount	0.997	0.995	1.000
LikeCount	0.999	0.999	0.999
IsCrawlable	0.275	0.102	0.744
IsAgeLimit	1.182	0.380	3.681
IsLiveContent	0.802	0.636	1.011
IsCommentsEnabled	1.115	0.901	1.381
IsAdsEnabled	0.719	0.599	0.862

Association of Predicted Probabilities and Observed Responses

Percent Concordant	61.0	Somers' D	0.222
Percent Discordant	38.9	Gamma	0.222
Percent Tied	0.1	Tau-a	0.002
Pairs	281571728	c	0.611

Effect	Point Estimate
ViewCount	1.000
DislikeCount	0.997

LikeCount	0.999
IsCrawlable	0.275
IsAgeLimit	1.182
IsLiveContent	0.802
IsCommentsEnabled	1.115
IsAdsEnabled	0.719



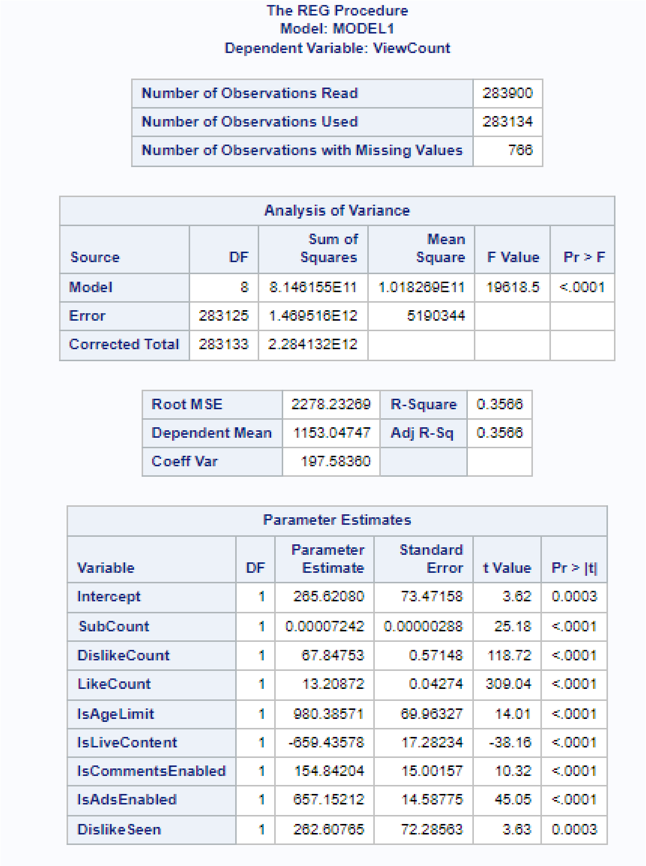
The only predictors that have parameter coefficient p-values less than the critical value of 0.05 are “DislikeCount”, “LikeCount”, “IsCrawlable”, and “IsAdsEnabled”. Knowing that the number of dislikes and likes are a main predictor if a video was published before or after the removal of the dislike count, we can continue the analysis.

The PROC CORR statement allowed “ViewCount” to be correlated with every variable to determine which variables should be included in the model.

Variable	**Correlation Coefficient**	**p-value**
UploadDate	-0.05258	< 0.0001
FetchedDate	-	-
SubCount	0.06471	< 0.0001
ViewCount	1.0000	-
LikeCount	0.55958	< 0.0001
DislikeCount	0.34649	< 0.0001
IsCrawlable	0.00022	0.9082
IsAgeLimit	0.02041	< 0.0001
IsLiveContent	-0.05293	< 0.0001

HasSubtitles	0.01466	< 0.0001
IsCommentsEnabled	0.02897	< 0.0001
IsAdsEnabled	0.13429	< 0.0001
year	-0.05253	< 0.0001
month	-0.00335	-0.00335
day	0.00164	0.00164
DislikeSeen	-0.01384	< 0.0001

The variables “UploadDate”, “FetchedDate”, “IsCrawlable”, “year”, “month”, and “day”, were not used for the model. 2 models were created with the remaining variables, one with and one without “DislikeSeen”. We are able to compare the two models to determine if the variable “DislikeSeen” has an impact on the number of views a video has.



The REG Procedure
Model: MODEL1
Dependent Variable: ViewCount

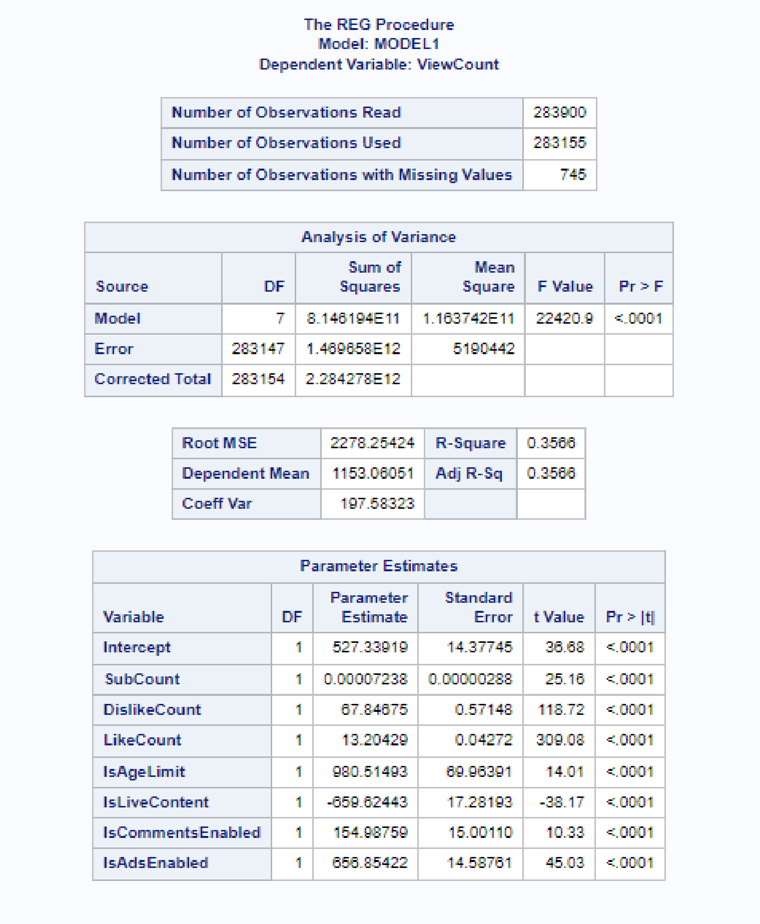
Number of Observations Read	283900
Number of Observations Used	283134
Number of Observations with Missing Values	766

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	8.146155E11	1.018269E11	19618.5	<.0001
Error	283125	1.469516E12	5190344		
Corrected Total	283133	2.284132E12			

Root MSE	2278.23269	R-Square	0.3566
Dependent Mean	1153.04747	Adj R-Sq	0.3566
Coeff Var	197.58360		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	265.62080	73.47158	3.62	0.0003
SubCount	1	0.00007242	0.00000288	25.18	<.0001
DislikeCount	1	67.84753	0.57148	118.72	<.0001
LikeCount	1	13.20872	0.04274	309.04	<.0001
IsAgeLimit	1	980.38571	69.96327	14.01	<.0001
IsLiveContent	1	-659.43578	17.28234	-38.16	<.0001
IsCommentsEnabled	1	154.84204	15.00157	10.32	<.0001
IsAdsEnabled	1	657.15212	14.58775	45.05	<.0001
DislikeSeen	1	262.60765	72.28563	3.63	0.0003

For the model that contains “DislikeSeen”, all of the parameter coefficients have p-values of less than the critical values of 0.05. The adjusted R^2 value is 0.3566 meaning that 35.66 percent of the variation in the model can be attributed to the independent variables. The F-value is quite large at 19618.5 with a p-value of less than 0.0001.



The REG Procedure
Model: MODEL1
Dependent Variable: ViewCount

Number of Observations Read	283900
Number of Observations Used	283155
Number of Observations with Missing Values	745

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	8.146194E11	1.163742E11	22420.9	<.0001
Error	283147	1.469658E12	5190442		
Corrected Total	283154	2.284278E12			

Root MSE	2278.25424	R-Square	0.3566
Dependent Mean	1153.06051	Adj R-Sq	0.3566
Coeff Var	197.58323		

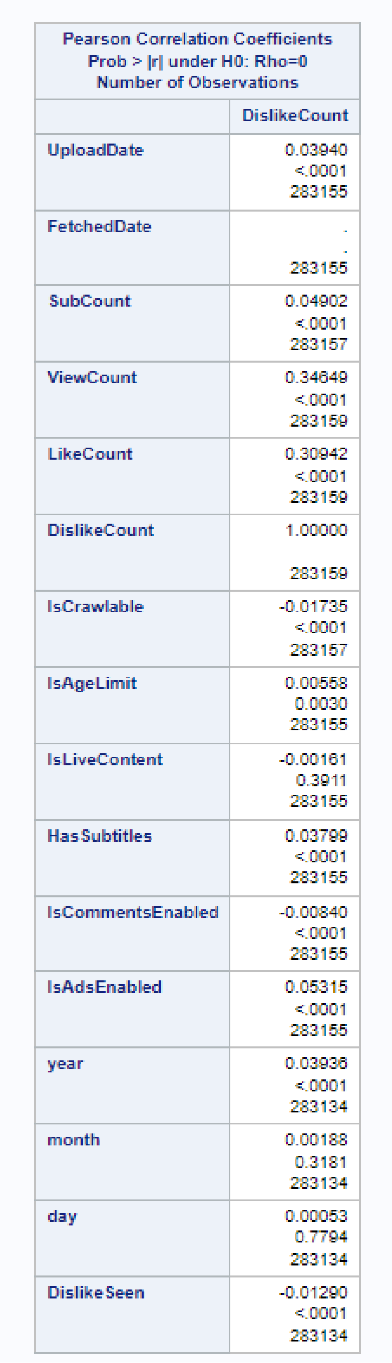
Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	527.33919	14.37745	36.68	<.0001
SubCount	1	0.00007238	0.00000288	25.16	<.0001
DislikeCount	1	67.84675	0.57148	118.72	<.0001
LikeCount	1	13.20429	0.04272	309.08	<.0001
IsAgeLimit	1	980.51493	69.96391	14.01	<.0001
IsLiveContent	1	-659.62443	17.28193	-38.17	<.0001
IsCommentsEnabled	1	154.98759	15.00110	10.33	<.0001
IsAdsEnabled	1	656.85422	14.58761	45.03	<.0001

For the model that does not contain “DislikeSeen”, all of the parameter coefficients have p-values of less than the critical values of 0.05. The adjusted R^2 value is also 0.3566 meaning that 35.66 percent of the variation in the model can be attributed to the independent variables. The F-value is even larger at 22420.9 with a p-value of less than 0.0001.

While many of the model’s utilities are the same, the F-value is larger for the one that does not contain “DislikeSeen”. There is not sufficient evidence to claim that the removal of the dislike count affects views on a video.

We can also use linear regression to determine if the dislike removal has affected the dislike count. Doing the same process for correlation, we get the table:

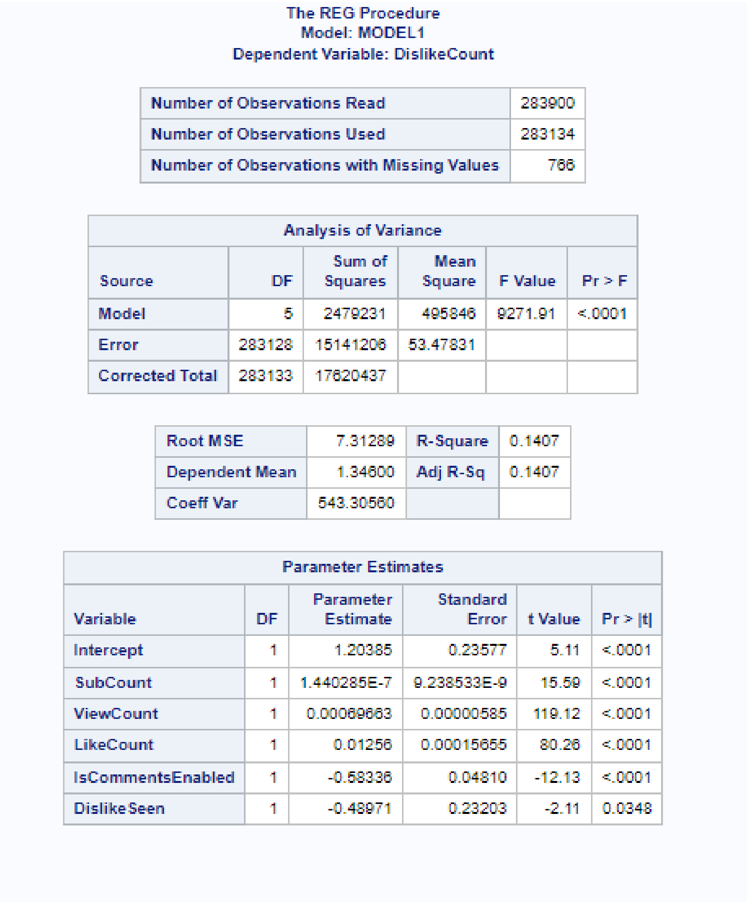


Pearson Correlation Coefficients
Prob > |r| under H0: Rho=0
Number of Observations

	DislikeCount
UploadDate	0.03940 <.0001 283155
FetchedDate	. . 283155
SubCount	0.04902 <.0001 283157
ViewCount	0.34649 <.0001 283159
LikeCount	0.30942 <.0001 283159
DislikeCount	1.00000 283159
IsCrawlable	-0.01735 <.0001 283157
IsAgeLimit	0.00558 0.0030 283155
IsLiveContent	-0.00161 0.3911 283155
HasSubtitles	0.03799 <.0001 283155
IsCommentsEnabled	-0.00840 <.0001 283155
IsAdsEnabled	0.05315 <.0001 283155
year	0.03936 <.0001 283134
month	0.00188 0.3181 283134
day	0.00053 0.7794 283134
DislikeSeen	-0.01290 <.0001 283134

The variables “UploadDate”, “FetchedDate”, “IsLiveContent”, “year”, “month”, and “day”, were not used for the model.

The original model had the parameter coefficients for “IsAgeLimit” and “IsLiveContent” have their p-values above the critical value of 0.05. Using a stepwise method, they were removed from the model to find a more effective model at predicting “DislikeCount”.



The REG Procedure
Model: MODEL1
Dependent Variable: DislikeCount

Number of Observations Read	283900
Number of Observations Used	283134
Number of Observations with Missing Values	766

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	2479231	495846	9271.91	<.0001
Error	283128	15141206	53.47831		
Corrected Total	283133	17620437			

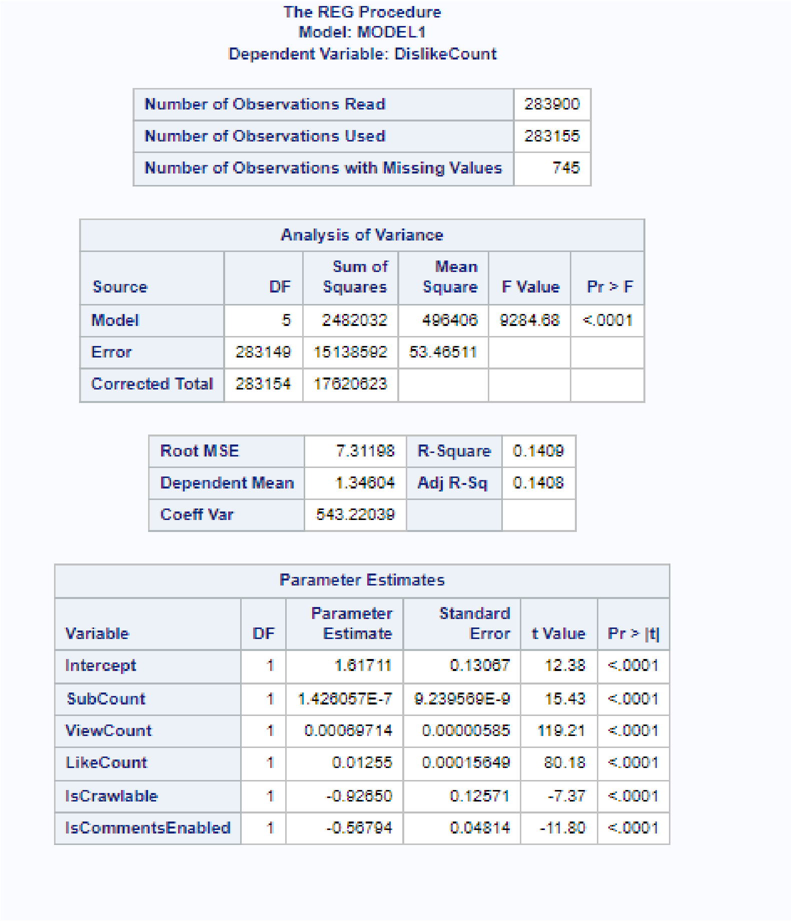
Root MSE	7.31289	R-Square	0.1407
Dependent Mean	1.34600	Adj R-Sq	0.1407
Coeff Var	543.30560		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.20385	0.23577	5.11	<.0001
SubCount	1	1.440285E-7	9.238533E-9	15.59	<.0001
ViewCount	1	0.00069663	0.00000585	119.12	<.0001
LikeCount	1	0.01256	0.00015655	80.26	<.0001
IsCommentsEnabled	1	-0.58336	0.04810	-12.13	<.0001
DislikeSeen	1	-0.48971	0.23203	-2.11	0.0348

For the model that contains “DislikeSeen”, all of the parameter coefficients have p-values of less than the critical values of 0.05. The adjusted R^2 value is also 0.1407 meaning that 14.07

percent of the variation in the model can be attributed to the independent variables. The F-value is at 9271.91 with a p-value of less than 0.0001.

For the model that does not contain “DislikeSeen”, parameter coefficients for “IsAgeLimit” and “IsAdsEnabled” have their p-values above the critical value of 0.05. Using a stepwise method, they were removed from the model to find a more effective model at predicting “DislikeCount”.



The REG Procedure
Model: MODEL1
Dependent Variable: DislikeCount

Number of Observations Read	283900
Number of Observations Used	283155
Number of Observations with Missing Values	745

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	2482032	496406	9284.68	<.0001
Error	283149	15138592	53.46511		
Corrected Total	283154	17620623			

Root MSE	7.31198	R-Square	0.1409
Dependent Mean	1.34604	Adj R-Sq	0.1408
Coeff Var	543.22039		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.61711	0.13067	12.38	<.0001
SubCount	1	1.426057E-7	9.239569E-9	15.43	<.0001
ViewCount	1	0.00069714	0.00000585	119.21	<.0001
LikeCount	1	0.01255	0.00015649	80.18	<.0001
IsCrawlable	1	-0.92650	0.12571	-7.37	<.0001
IsCommentsEnabled	1	-0.56794	0.04814	-11.80	<.0001

For the model that does not contain “DislikeSeen”, all of the parameter coefficients have p-values of less than the critical values of 0.05. The adjusted R^2 value is also 0.1408 meaning that 14.08 percent of the variation in the model can be attributed to the independent variables. The F-value is even larger at 9284.68 with a p-value of less than 0.0001.

Just like the linear regression models to predict “ViewCount”, many of the new model’s utility are the same, the F-value is larger for the one that does not contain “DislikeSeen”. There is not sufficient evidence to claim that the removal of the dislike count affects dislikes on a video.

2 Sample t-Test

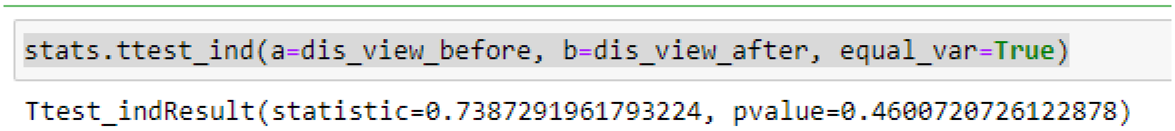
We will perform some 2 sample t-tests to determine whether or not the means for dislikes are equal or not before and after the removal of the dislike count. For the first statistical test, the metric dislikes/view was used for both before and after the removal of the dislike count as the amount of views might have an impact on dislikes.

Our hypothesis test is:

Ho: Mean for dislikes/views before = Mean for dislikes/views after

Ha: Mean for dislikes/views before ≠ Mean for dislikes/views after

After performing the statistical test, we get our results:



```
stats.ttest_ind(a=dis_view_before, b=dis_view_after, equal_var=True)
```

```
Ttest_indResult(statistic=0.7387291961793224, pvalue=0.4600720726122878)
```

The p-value for this test is 0.46007 thus there is not sufficient evidence to reject the null hypothesis and thus we cannot say that the means are not equal.

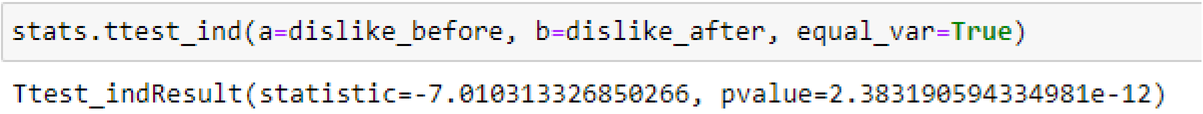
We perform another 2 sample t-test on the dislikes but we factor out the views and leave only the dislikes.

Our hypothesis test is:

Ho: Mean for dislikes before = Mean for dislikes after

Ha: Mean for dislikes before ≠ Mean for dislikes after

After performing the statistical test, we get our results:



```
stats.ttest_ind(a=dislike_before, b=dislike_after, equal_var=True)
```

```
Ttest_indResult(statistic=-7.010313326850266, pvalue=2.383190594334981e-12)
```

The p-value for this test is very small at 2.38 e-12.With this statistic, there is sufficient evidence to reject the null hypothesis and accept the alternative hypothesis that the means are not equal.

Ethical Recommendations

An exploration of YouTube video metadata allows us to analyze the intentions behind certain content creators and the reactions of the viewers for the videos. Content creators use the video streaming platform as a way for income for themselves and sometimes their companies; with so many users on the platform the only way to stand out seems to not be the most ethical decision.

The titles of YouTuber videos are supposed to draw attention to the video while informing the viewer what the video is supposed to be about. Many YouTubers take advantage of this by making the titles of the videos misleading, also known as clickbait. They trick the viewer into clicking on the video so their video can gain a view. With more views on their video, the video is placed onto more viewer's recommended pages where the cycle repeats. Higher views on videos will lead to more revenue for the content creator, a way to make money by deceiving the viewers.

It is very difficult to identify if a video is clickbait or not based on the title alone, one would have to actually watch the video on the subject matter to determine the answer. With this idea, having an algorithm to determine if a video is clickbait or not is near impossible, which is why the prediction of clickbait in the analysis is not perfect. The algorithm is meant to identify which videos have the potential to be clickbait based on the metadata, especially the titles. Of course there is the idea that a video is not clickbait even though it fits the criteria for potentially clickbait. Keeping that idea in mind, it appears that an honest YouTuber can still grow larger in popularity if they still use those keywords in their titles.

YouTube plays a huge role in modern day culture, many people use it as a main form of entertainment rather than watching regular movies or TV series, and if regular YouTube is not enough, YouTube even has a service where one can buy or rent movies, or even subscribe to their own TV service). YouTube’s services are plentiful and allow the users to enjoy many sources of entertainment all on one platform. It is important that the users of this platform understand the importance of removing clickbait. Without the removal of clickbait on the platform, no video can be trusted. The platform will be a mess of truth and lies and no one will know the difference between the two because the only thing the content creators would care about is the revenue they obtain from easy views. Views should be earned with effort and creativity which can be seen by watching the video, not by tricking viewers into clicking on a video that gives false promises or fake information. The rise of fake news is already a problem in current media, bringing said falsities into an entertainment platform will discredit not only important topics, but even the minute and mundane ones.

Challenges

There were a few challenges that were encountered during this analysis. First, the initial data that was explored was not distributed normally at all. While this might be normal, it was difficult to work with data that had many observations with lower values but several observations with values that are tens of thousands times the mean of the data; being able to find patterns in the data had to be done by filtering the outliers out of the data.

Another issue was the size of the data was quite difficult to manage; the whole of the data contained almost 300,000 observations. Filtering down the data, such as performing an operation with textcat to identify which language each video title is in took over 10 hours to complete. Many worries were in place regarding if the operation in Jupyter Hub would even work or if Google Chrome/Jupyter Hub would crash as the data set was too large to work with. Fortunately, it worked on the first attempt. Working with textcat on a data set this large again would require either the code or textcat to be more efficient so it will not have to take nearly as much time as it did for this project.

Additionally, textcat posed a problem at first because even though it looked like textcat correctly identified the language the video title was in, filtering out the non-English titles did not work as intended because there were still non-English titles in the mix. The code was combed thoroughly and fixed when needed so that the python cells did exactly what was needed to fix the issue. Several of the cells needed to be fixed when it came to the “for loops” after textcat was used which makes sense as the tables displayed immediately when textcat was finished working showed the correct result but tokenizing all the titles appeared wrong.

Another issue that was faced during this project was finding a way to classify videos to use the Naive Bayes Classification method. The data did not include if a video was clickbait or not (as there is currently no way to do so) so creative thinking had to be put in place in order to carry forwards with this part of the analysis. At first the standard deviation for the ViewCount and SubCount variables were to be used to identify if a video was potentially clickbait but it was discovered that the standard deviation was greater than the mean as the data was heavily skewed right for these particular variables. The solution to this problem was to use the median instead of the mean as it was a more effective measure for the ViewCount and SubCount.

Recommendations

Overall I believe that the analysis is a great starting point for continuing the examination of the effect titles in YouTube videos on metadata. While YouTube does not currently have a method to detect clickbait on their interface, it would be extremely beneficial to the community as a whole as viewers can save their efforts and not click on irrelevant, misleading videos and content creators can make better quality videos that stay true to their content rather than rely on stretching the truth for “easy views”. Being able to predict if a video is potentially clickbait is a start to cleaning the platform of dishonesty and will allow it to return to the values it has held for over a decade.

Additionally, it was intriguing to look into viewer’s habits regarding the dislike count. While there was not much data to observe after the removal of the dislike count, it still seems like there is a difference in dislikes before and after the removal. The video publish dates only go to early 2022, meaning that there has been many months since then that could assist in researching this question further. With more data, it can confirm our results or possibly shed light onto other possibilities that were never even considered.

If this project was continued, languages other than English would be researched to see if many of the phenomena witnessed in the data researched expands outside of the main language on the platform. Out of the almost 300,000 observations, a little more than a third were used for the Naive Bayes Classification as they were in English. While 100,000 observations is a great amount of data to work with, it is only but a small fraction of the total amount of videos that actually exist on the streaming platform, and only a portion of the videos that are a part of the English language. The initial data is only a part of the collective as a whole; over 4.5 billion videos could have been used for the analysis but due to computational limitations, only a small part was used. Further research can use the other parts of the data not downloaded to confirm or deny the results obtained in the analysis.

The third research question that was not pursued in the analysis can be continued now that the video titles have been ranked. The views of a video are a good indicator if the video is clickbait or not and the likes/dislikes of a video can also be an effective way to determine if a video is clickbait. Additionally, what the likes/dislikes tells us compared to the views is if the viewers are actively fighting against clickbait themselves. Theoretically, if a video that is marked as potentially clickbait has many likes, there is a greater chance that it is actually not clickbait relative to a potentially clickbait video that has many dislikes on it. If the users are taking initiative, then they should dislike the video. This is interesting when combined with research question 2 as perhaps the viewers will not dislike the video if they do not see the dislike count even if it is clickbait. A further study on this matter is intriguing as it mixes together research question 1 and 2, this research question is a great continuation since the other research questions have been studied.

More research can be done on variables that were not used in the data such as comments. A research question that could be posed is if video titles have an effect on the comments of a video, if they are more positive or negative. While views, likes, etc are a great measure if a video is considered “good”, the comments are actual inputs by the users which can be dissected for more precise opinions compared to a simple like/dislike. Another consideration is the idea of the video thumbnail. This is a challenging one as it will take image processing but the image a user sees before deciding whether to click on a video or not heavily determines whether they actually do so; it is a huge factor that was not considered in this analysis.

References

YouTube Revenue and Usage Statistics (2022). Business of Apps. (2022, September 6). Retrieved

September 4, 2022, from https://www.businessofapps.com/data/youtube-statistics

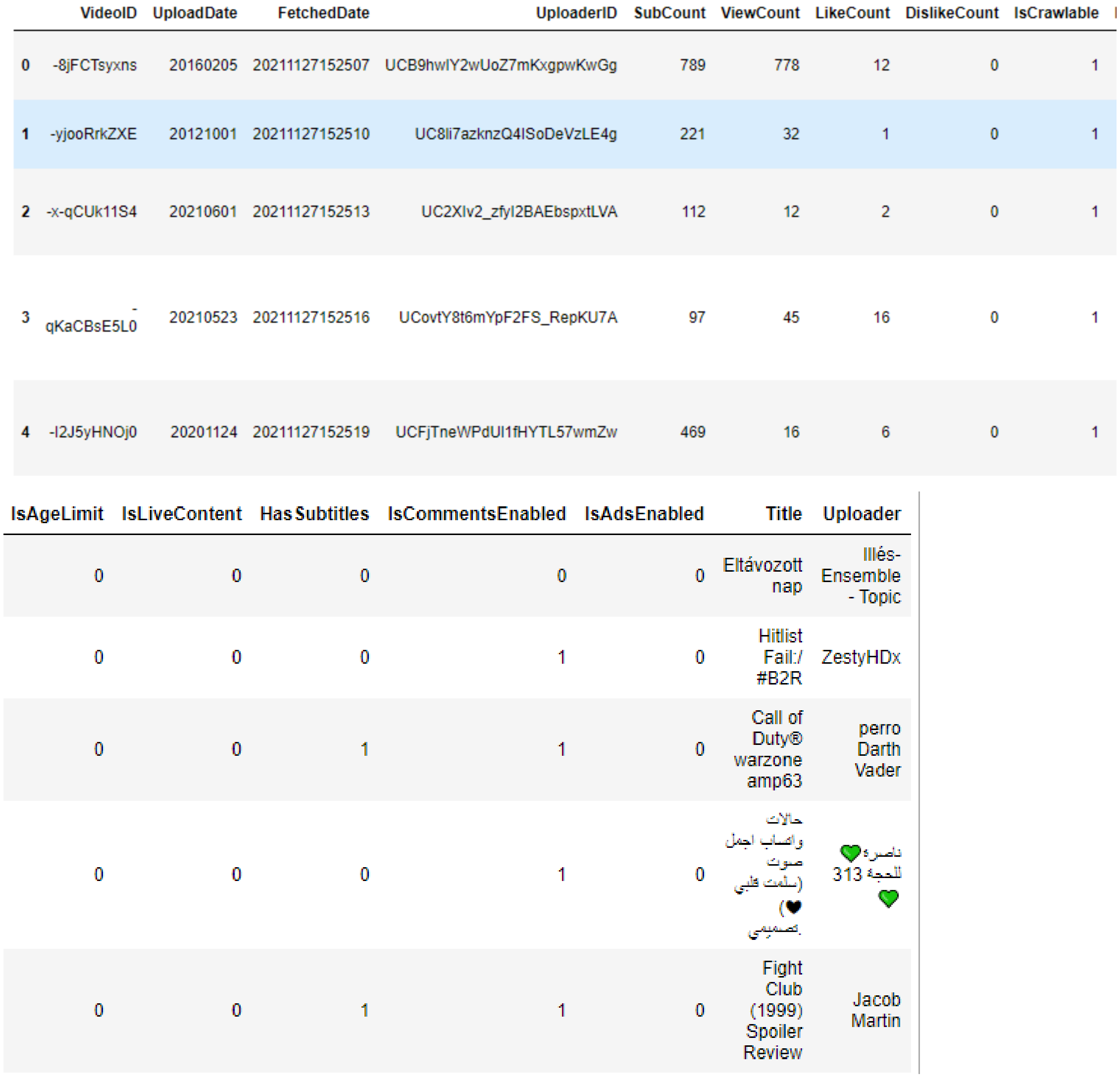
Rui, L. T., Afif, Z. A., Saedudin, R., Mustapha, A., &amp; Razali, N. (2019). A Regression Approach for Prediction of Youtube Views. Bulletin of Electrical Engineering and Informatics.

Braun, P., Cuzzrocrea, A., Kim, S., Leung, C., Francisco, J., Matundan, A., &amp; Singha, R. R. (2017). Enhanced Prediction of User-Preferred YouTube Videos Based on Cleaned Viewing Pattern History. ScienceDirect.

Arthurs, J., Drakopoulou, S., &amp; Gandini, A. (2018). Researching youtube. Convergence: The International Journal of Research into New Media Technologies, 24(1), 3–15. https://doi.org/10.1177/1354856517737222

YouTube executive team | comparably. (n.d.). Retrieved September 4, 2022, from https://www.comparably.com/companies/youtube/executive-team

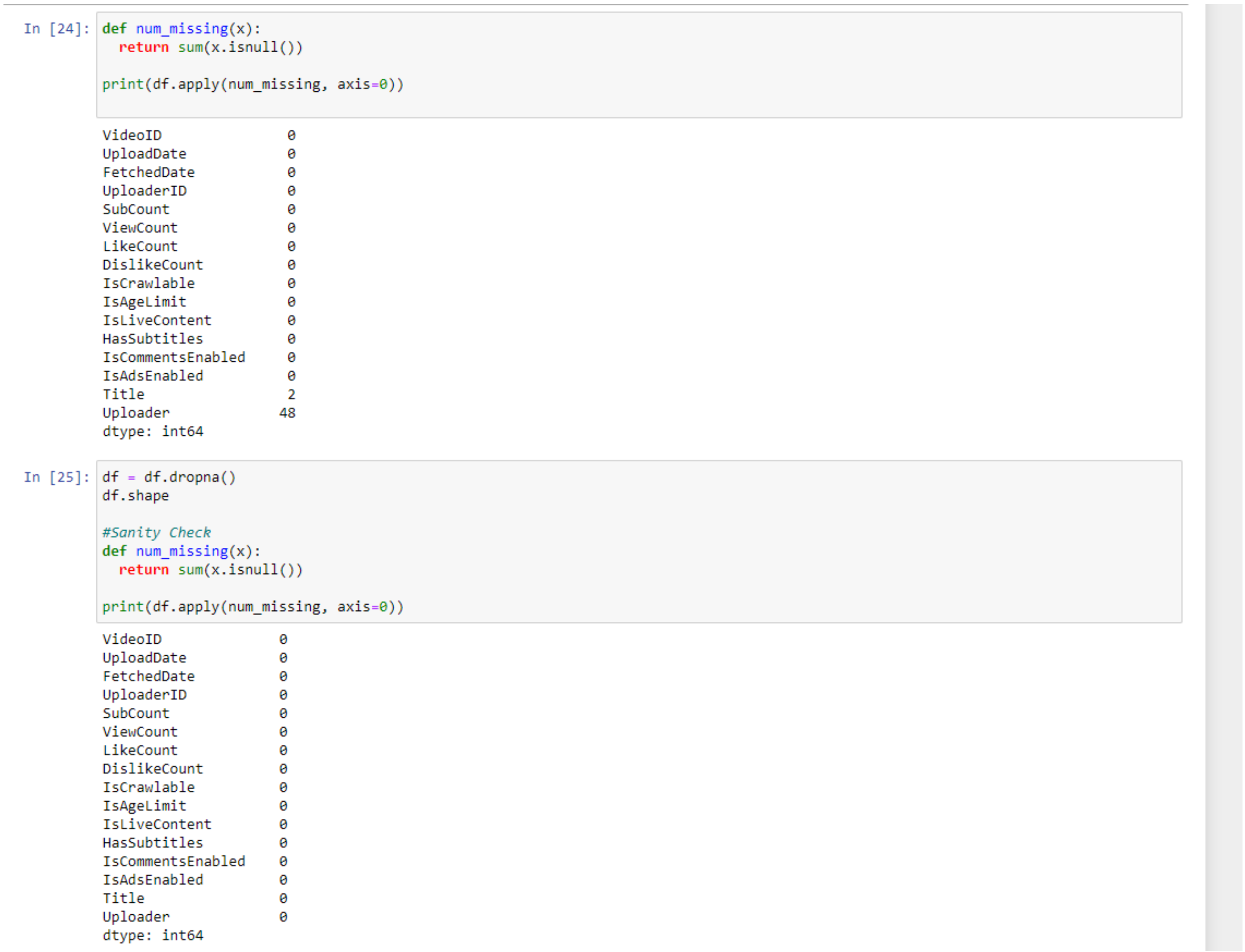
Appendix



	VideoID	UploadDate	FetchedDate	UploaderID	SubCount	ViewCount	LikeCount	DislikeCount	IsCrawlable
0	-8jFCTsyxns	20160205	20211127152507	UCB9hwlY2wUoZ7mKxgpwKwGg	789	778	12	0	1
1	-yjooRrkZXE	20121001	20211127152510	UC8li7azknzQ4lSoDeVzLE4g	221	32	1	0	1
2	-x-qCUk11S4	20210601	20211127152513	UC2XIv2_zfyI2BAEbspxtLVA	112	12	2	0	1
3	-qKaCBsE5L0	20210523	20211127152516	UCovtY8t6mYpF2FS_RepKU7A	97	45	16	0	1
4	-I2J5yHNOj0	20201124	20211127152519	UCFjTneWPdUl1fHYTL57wmZw	469	16	6	0	1

IsAgeLimit	IsLiveContent	HasSubtitles	IsCommentsEnabled	IsAdsEnabled	Title	Uploader
0	0	0	0	0	Eltávozott nap	Illés-Ensemble - Topic
0	0	0	1	0	Hitlist Fail:/ #B2R	ZestyHDx
0	0	1	1	0	Call of Duty® warzone amp63	perro Darth Vader
0	0	0	1	0	حالات واتساب اجمل صوت (سلمت قلبي ♥) تصميمي.	ناصرة 💚 للحجة 313 💚
0	0	1	1	0	Fight Club (1999) Spoiler Review	Jacob Martin

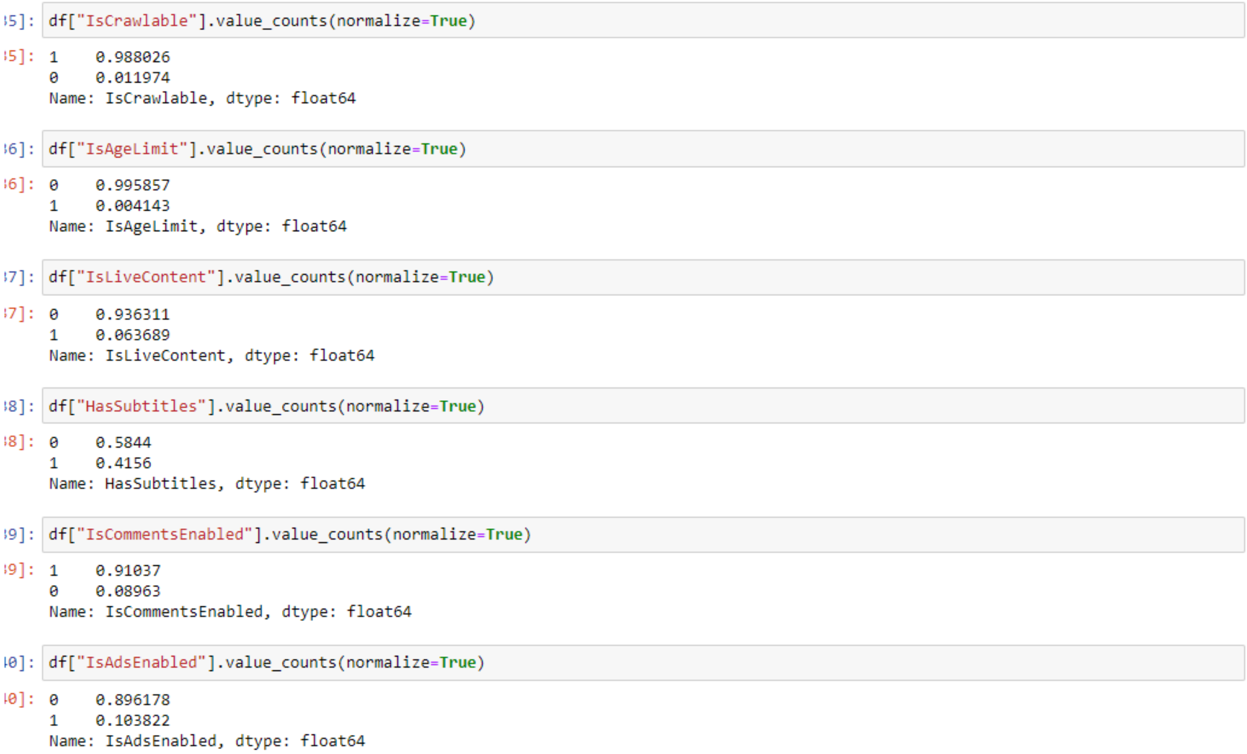
Figure 1



```
In [24]: def num_missing(x):
           return sum(x.isnull())

         print(df.apply(num_missing, axis=0))

VideoID              0
UploadDate           0
FetchedDate          0
UploaderID           0
SubCount             0
ViewCount            0
LikeCount            0
DislikeCount         0
IsCrawlable          0
IsAgeLimit           0
IsLiveContent        0
HasSubtitles         0
IsCommentsEnabled    0
IsAdsEnabled         0
Title                2
Uploader            48
dtype: int64
```

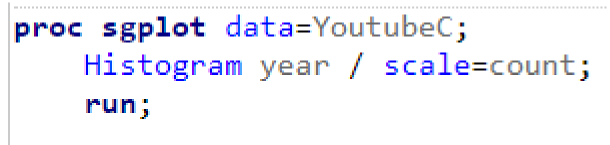
```
In [25]: df = df.dropna()
         df.shape

         #Sanity Check
         def num_missing(x):
           return sum(x.isnull())

         print(df.apply(num_missing, axis=0))

VideoID             0
UploadDate          0
FetchedDate         0
UploaderID          0
SubCount            0
ViewCount           0
LikeCount           0
DislikeCount        0
IsCrawlable         0
IsAgeLimit          0
IsLiveContent       0
HasSubtitles        0
IsCommentsEnabled   0
IsAdsEnabled        0
Title               0
Uploader            0
dtype: int64
```

Figure 2

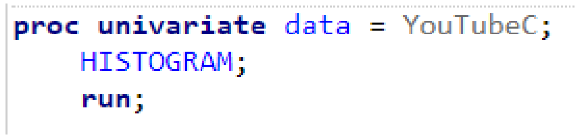


```
5]: df["IsCrawlable"].value_counts(normalize=True)

5]: 1    0.988026
    0    0.011974
    Name: IsCrawlable, dtype: float64

6]: df["IsAgeLimit"].value_counts(normalize=True)

6]: 0    0.995857
    1    0.004143
    Name: IsAgeLimit, dtype: float64

7]: df["IsLiveContent"].value_counts(normalize=True)

7]: 0    0.936311
    1    0.063689
    Name: IsLiveContent, dtype: float64

8]: df["HasSubtitles"].value_counts(normalize=True)

8]: 0    0.5844
    1    0.4156
    Name: HasSubtitles, dtype: float64

9]: df["IsCommentsEnabled"].value_counts(normalize=True)

9]: 1    0.91037
    0    0.08963
    Name: IsCommentsEnabled, dtype: float64

0]: df["IsAdsEnabled"].value_counts(normalize=True)

0]: 0    0.896178
    1    0.103822
    Name: IsAdsEnabled, dtype: float64
```

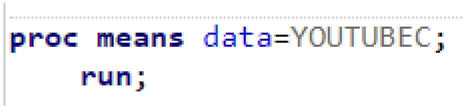
Figure 3



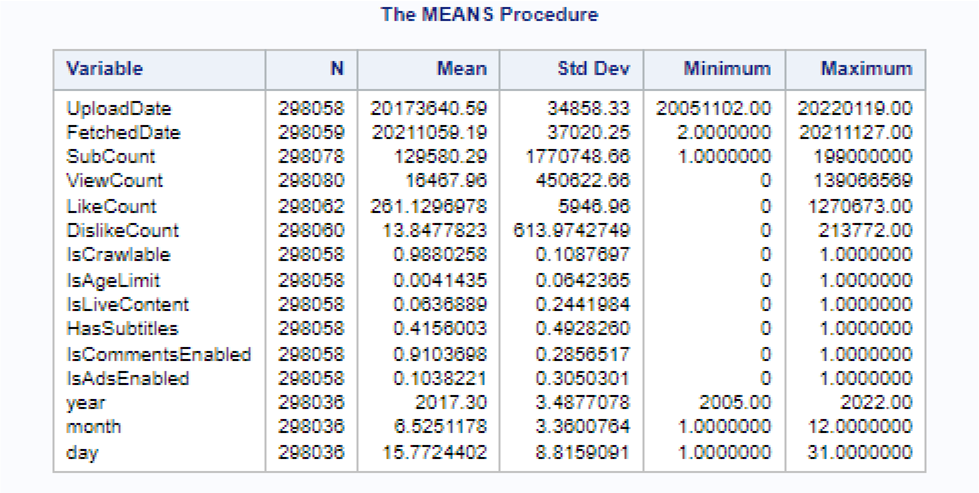
```
proc sgplot data=YoutubeC;
    Histogram year / scale=count;
    run;
```



```
proc univariate data = YouTubeC;
    HISTOGRAM;
    run;
```



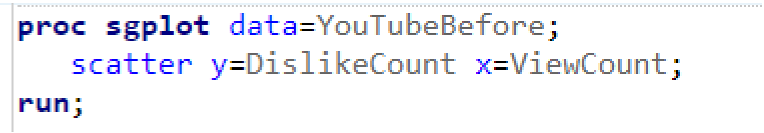
```
proc means data=YOUTUBEC;
    run;
```



The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
UploadDate	298058	20173640.59	34858.33	20051102.00	20220119.00
FetchedDate	298059	20211059.19	37020.25	2.0000000	20211127.00
SubCount	298078	129580.29	1770748.66	1.0000000	199000000
ViewCount	298080	16467.96	450622.66	0	139066569
LikeCount	298062	261.1296978	5946.96	0	1270673.00
DislikeCount	298060	13.8477823	613.9742749	0	213772.00
IsCrawlable	298058	0.9880258	0.1087697	0	1.0000000
IsAgeLimit	298058	0.0041435	0.0642365	0	1.0000000
IsLiveContent	298058	0.0636889	0.2441984	0	1.0000000
HasSubtitles	298058	0.4156003	0.4928260	0	1.0000000
IsCommentsEnabled	298058	0.9103698	0.2856517	0	1.0000000
IsAdsEnabled	298058	0.1038221	0.3050301	0	1.0000000
year	298036	2017.30	3.4877078	2005.00	2022.00
month	298036	6.5251178	3.3600764	1.0000000	12.0000000
day	298036	15.7724402	8.8159091	1.0000000	31.0000000

Figure 4



```
proc sgplot data=YouTubeBefore;
   scatter y=DislikeCount x=ViewCount;
run;
```

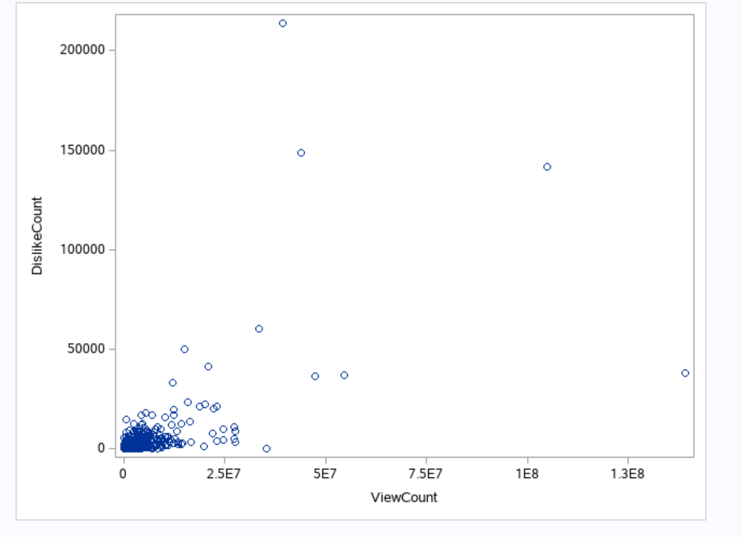
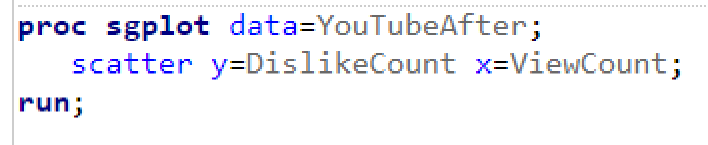


Figure 5



```
proc sgplot data=YouTubeAfter;
   scatter y=DislikeCount x=ViewCount;
run;
```

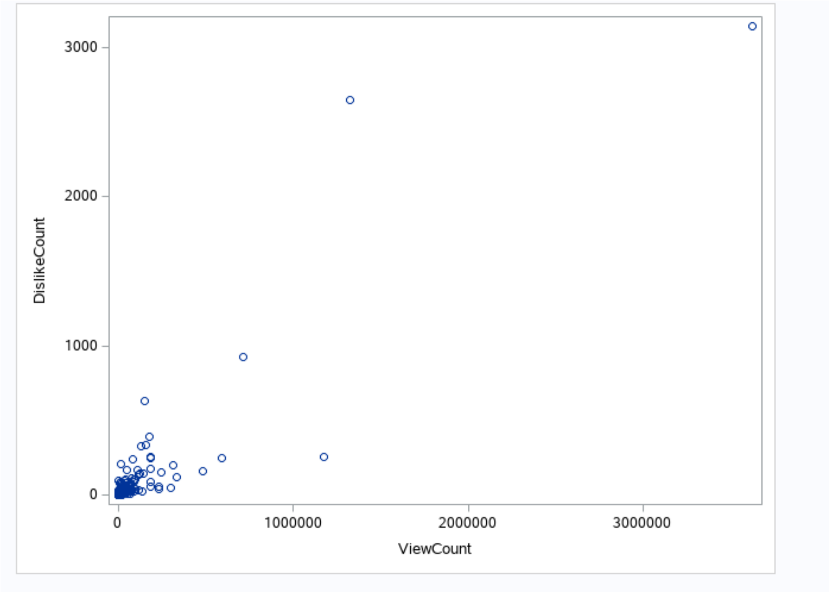
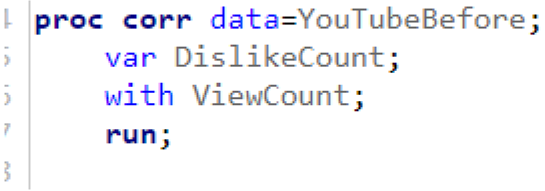
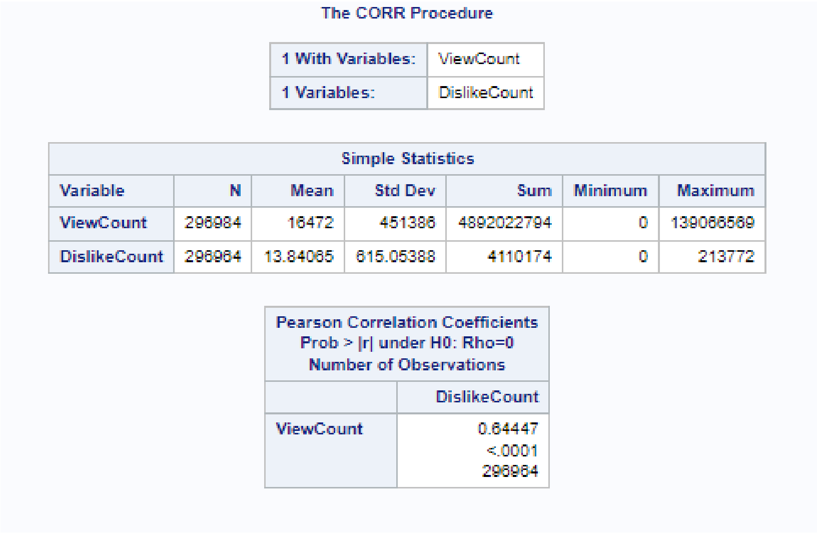


Figure 6



```
proc corr data=YouTubeBefore;
    var DislikeCount;
    with ViewCount;
    run;
```



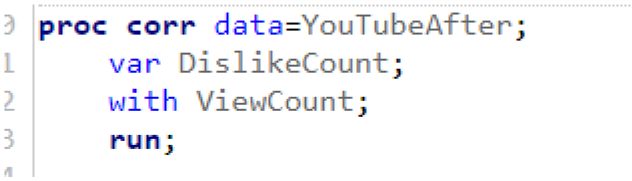
The CORR Procedure

1 With Variables:	ViewCount
1 Variables:	DislikeCount

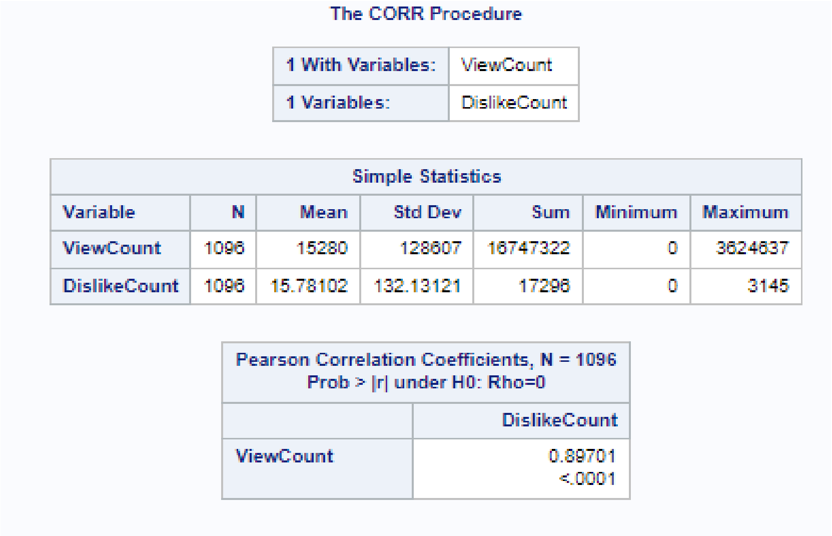
Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
ViewCount	296984	16472	451386	4892022794	0	139066569
DislikeCount	296984	13.84065	615.05388	4110174	0	213772

Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations	
	DislikeCount
ViewCount	0.64447 <.0001 296984

Figure 7



```
proc corr data=YouTubeAfter;
    var DislikeCount;
    with ViewCount;
    run;
```



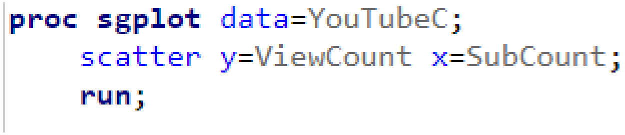
The CORR Procedure

1 With Variables:	ViewCount
1 Variables:	DislikeCount

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
ViewCount	1096	15280	128607	16747322	0	3624637
DislikeCount	1096	15.78102	132.13121	17296	0	3145

Pearson Correlation Coefficients, N = 1096 Prob > |r| under H0: Rho=0	
	DislikeCount
ViewCount	0.89701 <.0001

Figure 8



```
proc sgplot data=YouTubeC;
    scatter y=ViewCount x=SubCount;
    run;
```

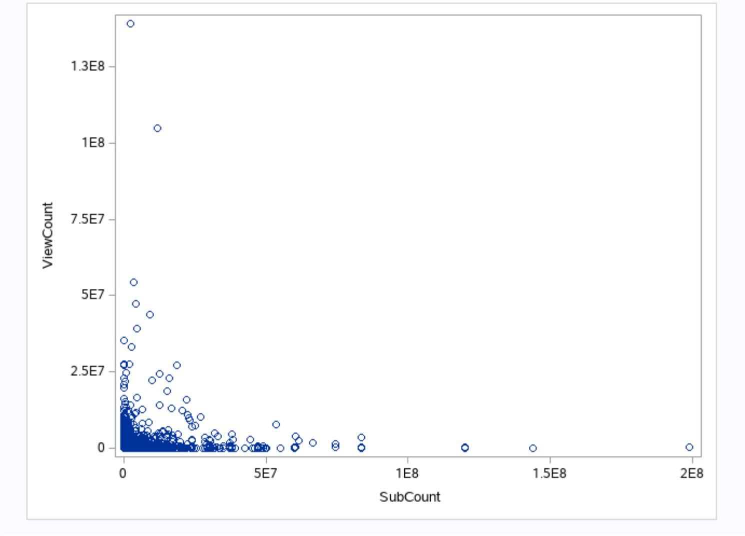
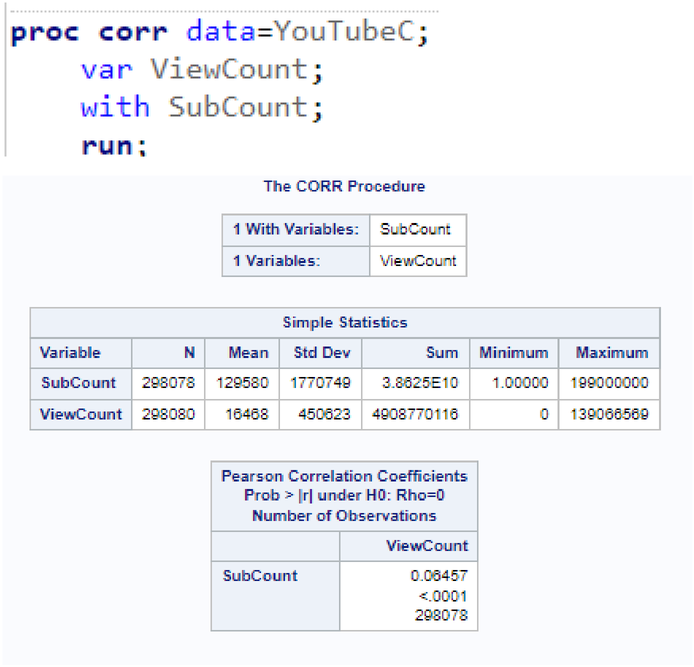


Figure 9



```
proc corr data=YouTubeC;
    var ViewCount;
    with SubCount;
    run;
```

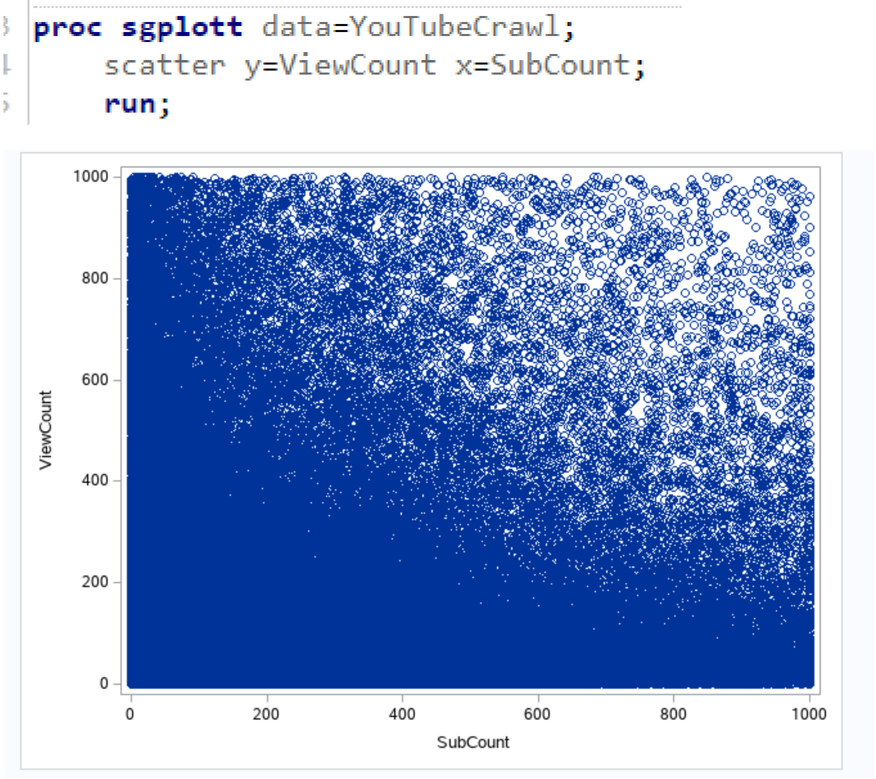
The CORR Procedure

1 With Variables:	SubCount
1 Variables:	ViewCount

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
SubCount	298078	129580	1770749	3.8625E10	1.00000	199000000
ViewCount	298080	16468	450623	4908770116	0	139066569

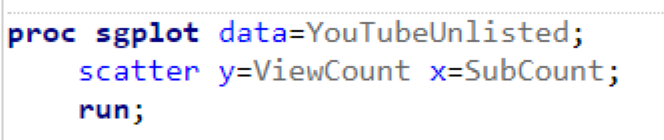
Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations	
	ViewCount
SubCount	0.06457 <.0001 298078

Figure 10



```
proc sgplott data=YouTubeCrawl;
    scatter y=ViewCount x=SubCount;
    run;
```

Figure 11



```
proc sgplot data=YouTubeUnlisted;
    scatter y=ViewCount x=SubCount;
    run;
```

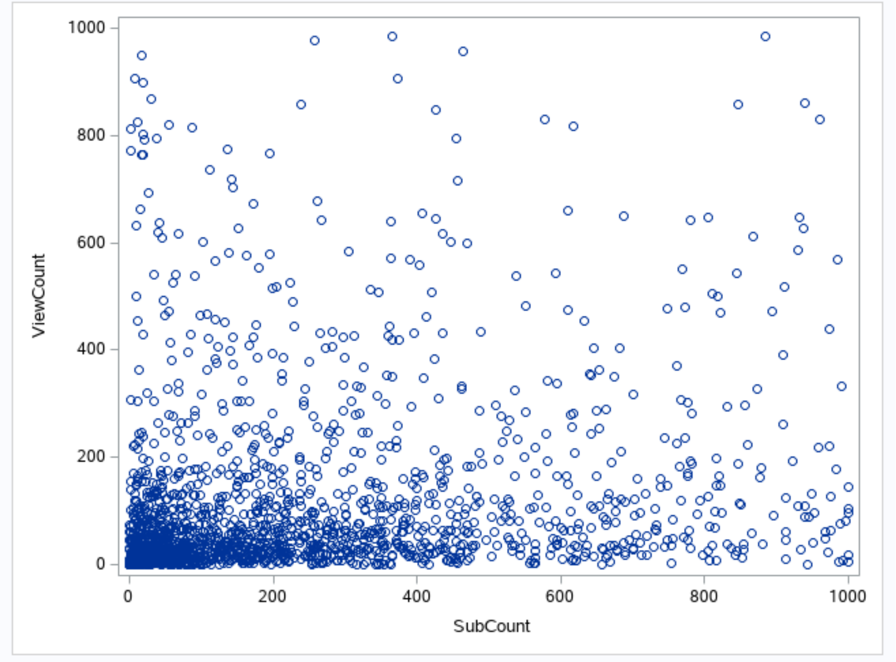
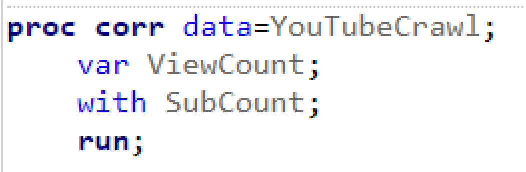
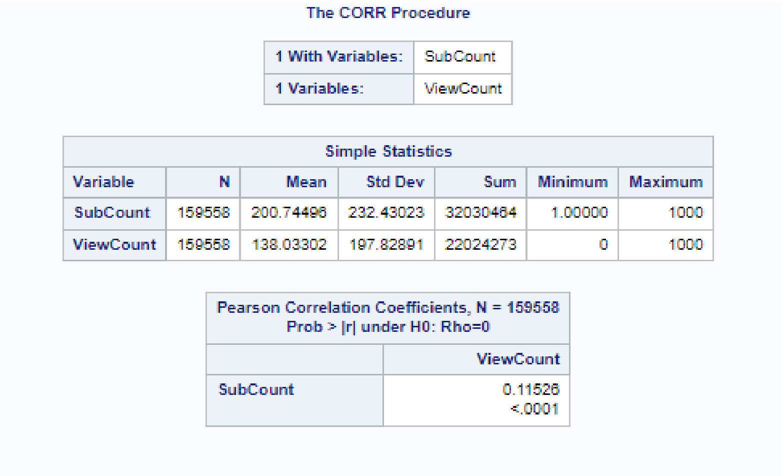


Figure 12

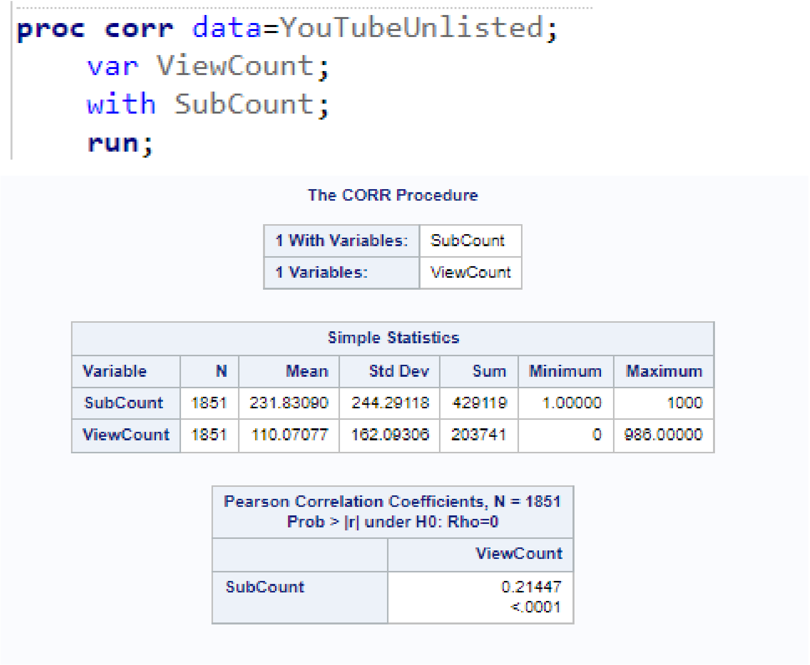


```
proc corr data=YouTubeCrawl;
    var ViewCount;
    with SubCount;
    run;
```



```
                         The CORR Procedure

                   1 With Variables:   SubCount
                   1 Variables:        ViewCount

                          Simple Statistics
Variable       N        Mean     Std Dev        Sum    Minimum   Maximum
SubCount   159558   200.74496   232.43023   32030464   1.00000      1000
ViewCount  159558   138.03302   197.82891   22024273         0      1000

            Pearson Correlation Coefficients, N = 159558
                      Prob > |r| under H0: Rho=0
                                              ViewCount
            SubCount                            0.11526
                                                 <.0001
```

Figure 13



```
proc corr data=YouTubeUnlisted;
    var ViewCount;
    with SubCount;
    run;
```

The CORR Procedure

1 With Variables:	SubCount
1 Variables:	ViewCount

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
SubCount	1851	231.83090	244.29118	429119	1.00000	1000
ViewCount	1851	110.07077	162.09306	203741	0	986.00000

Pearson Correlation Coefficients, N = 1851
Prob > \|r\| under H0: Rho=0

	ViewCount
SubCount	0.21447 <.0001

Figure 14

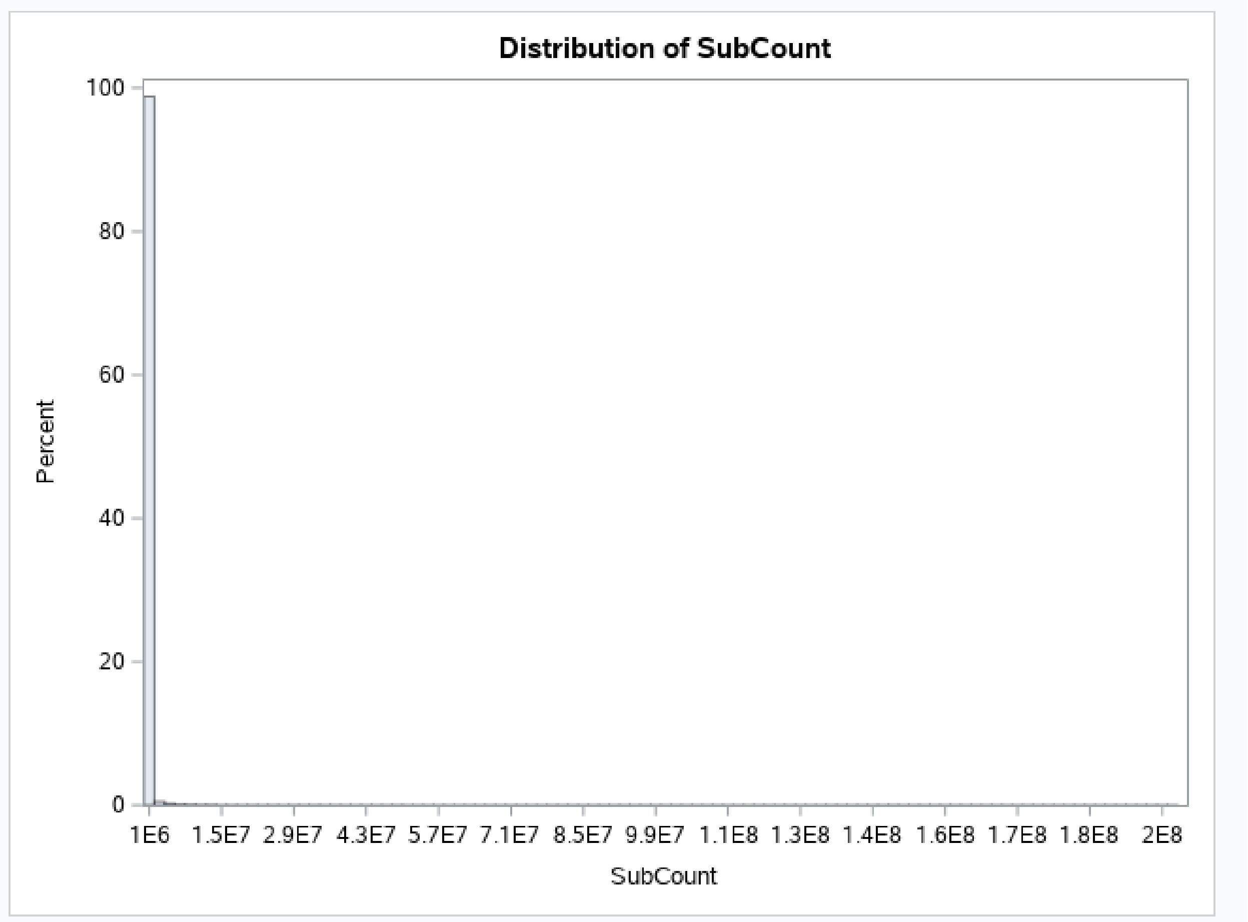

Distribution of SubCount
Percent
100
80
60
40
20
0
1E6
1.5E7
2.9E7
4.3E7
5.7E7
7.1E7
8.5E7
9.9E7
1.1E8
1.3E8
1.4E8
1.6E8
1.7E8
1.8E8
2E8
SubCount

Figure 15

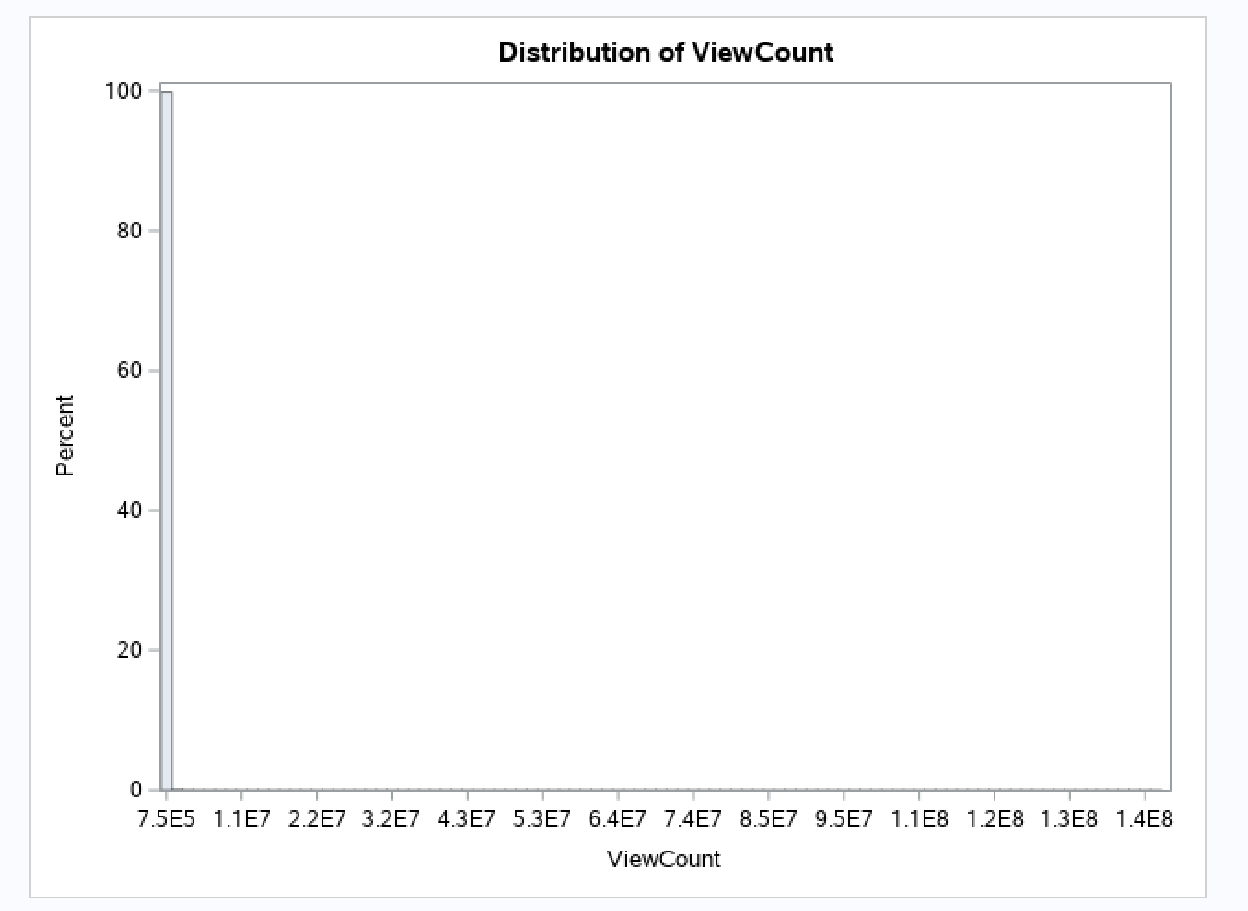

Distribution of ViewCount
100
80
60
40
20
0
Percent
7.5E5
1.1E7
2.2E7
3.2E7
4.3E7
5.3E7
6.4E7
7.4E7
8.5E7
9.5E7
1.1E8
1.2E8
1.3E8
1.4E8
ViewCount

Figure 16

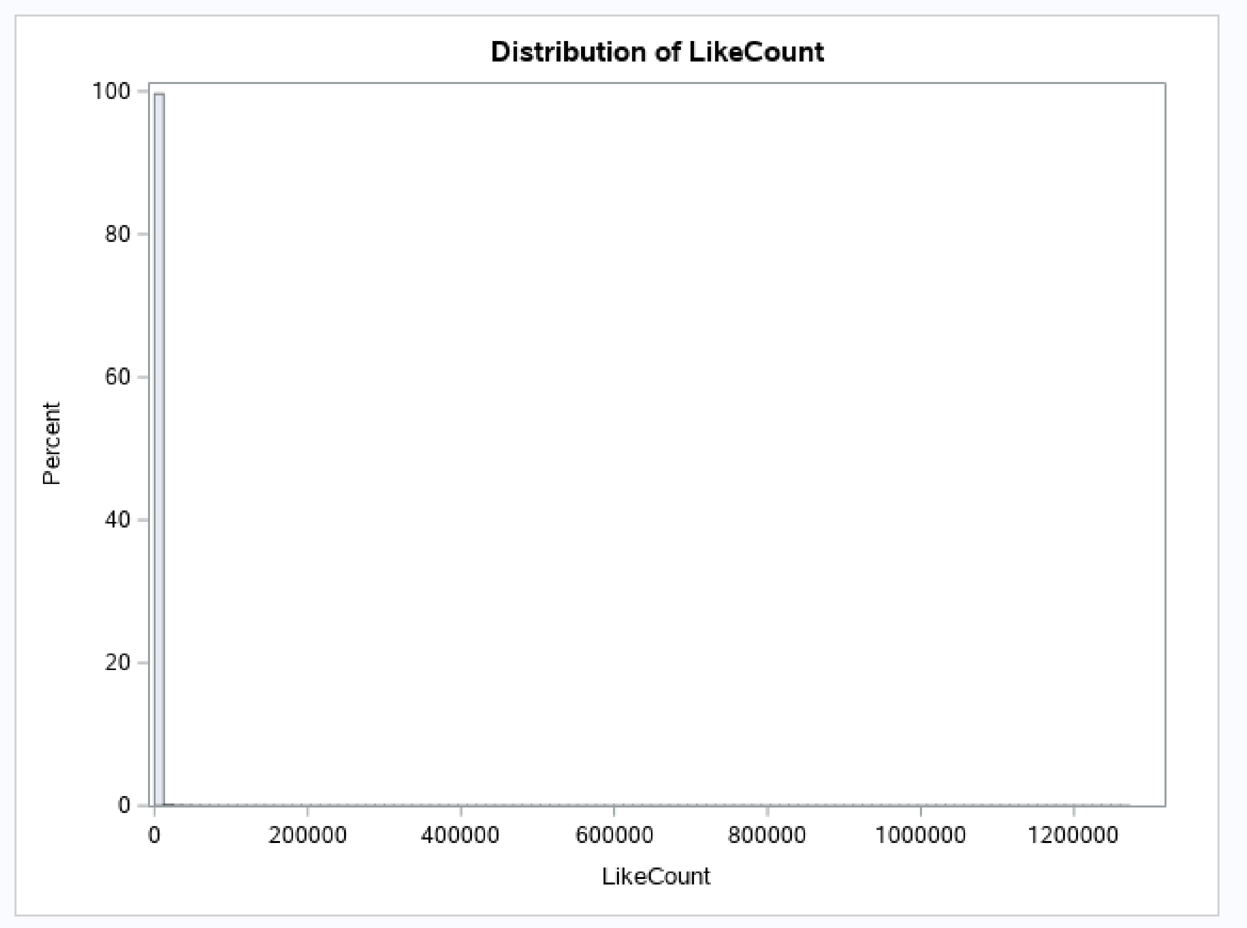

Distribution of LikeCount
Percent
100
80
60
40
20
0
0
200000
400000
600000
800000
1000000
1200000
LikeCount

Figure 17

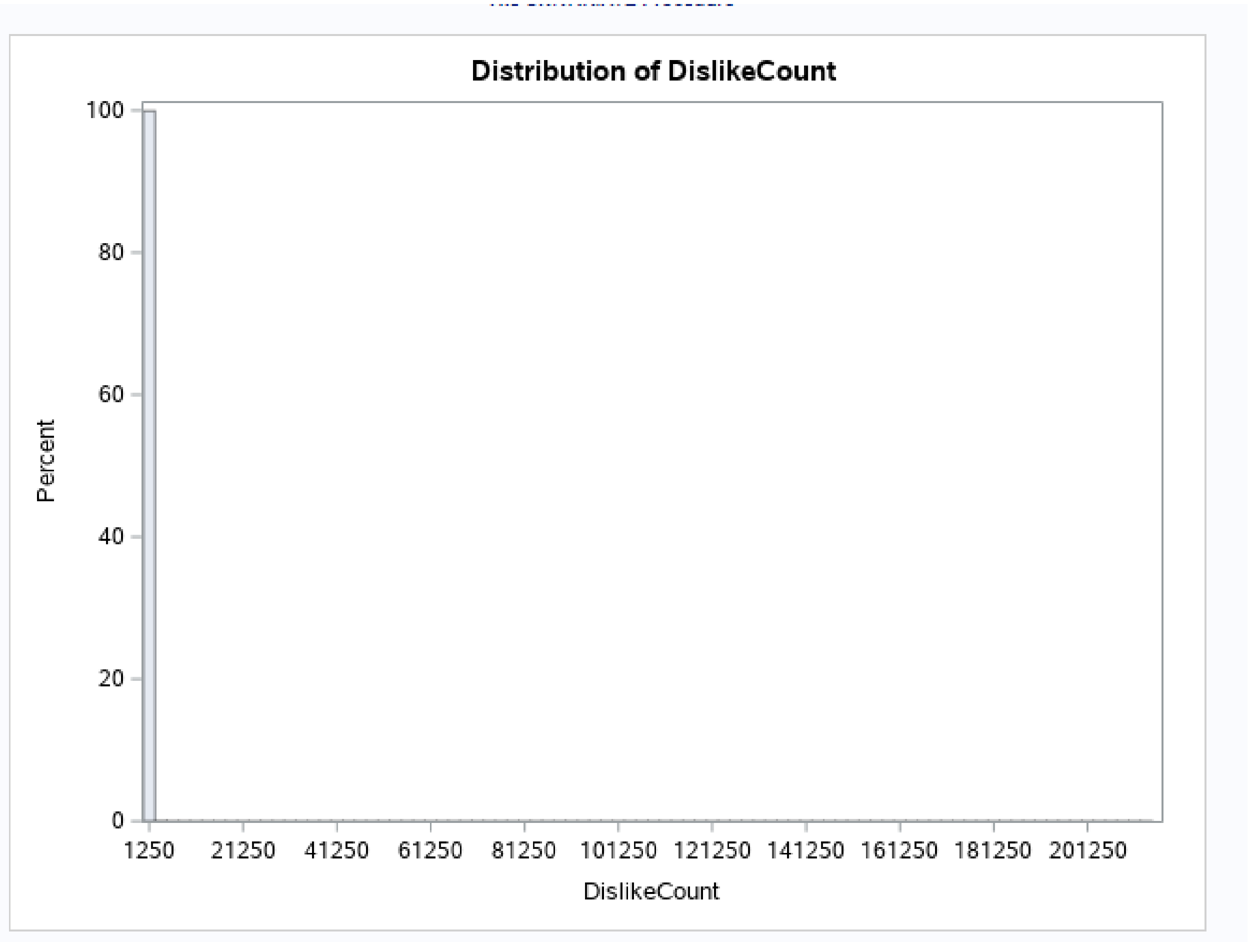

Distribution of DislikeCount
Percent
100
80
60
40
20
0
1250
21250
41250
61250
81250
101250
121250
141250
161250
181250
201250
DislikeCount

Figure 18

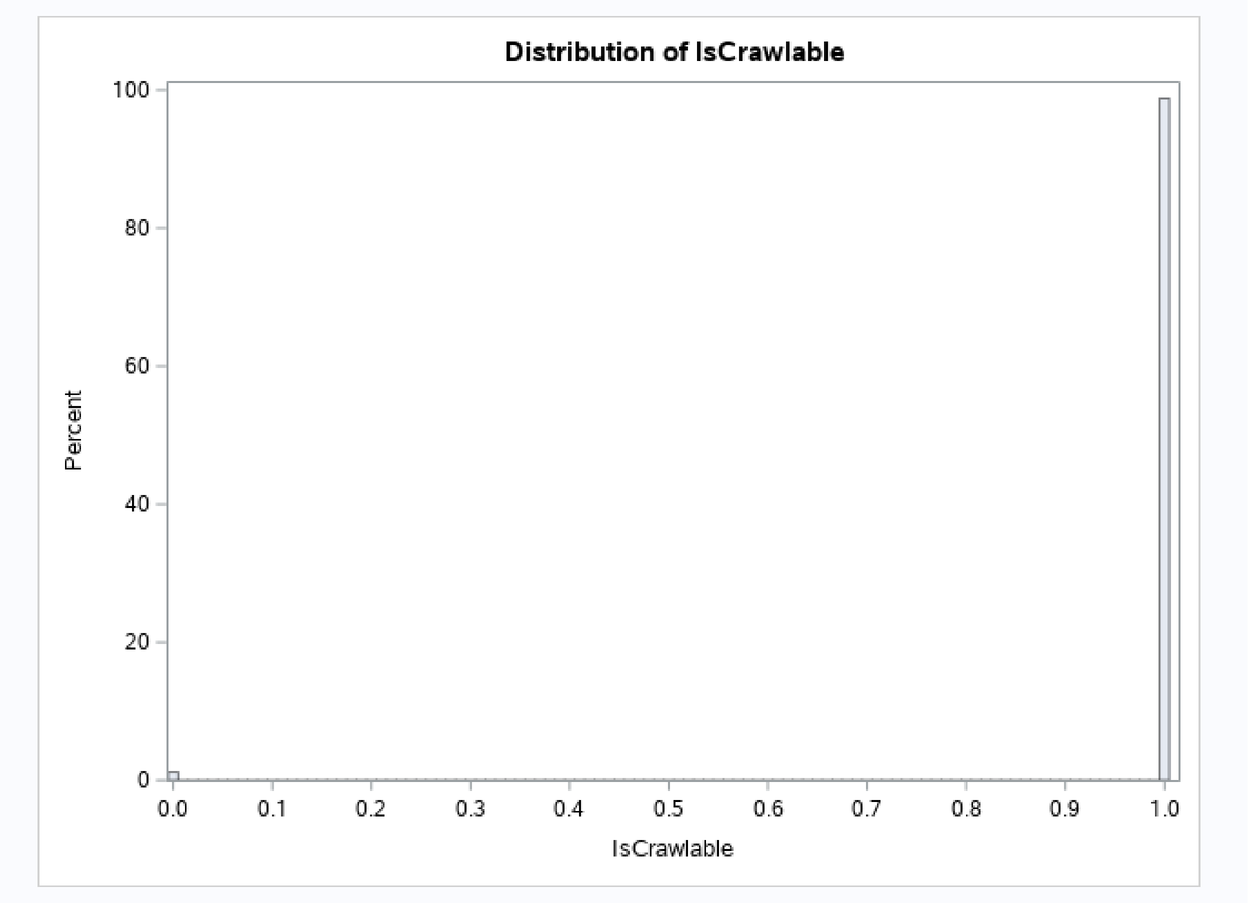

Distribution of IsCrawlable
Percent
100
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
IsCrawlable

Figure 19

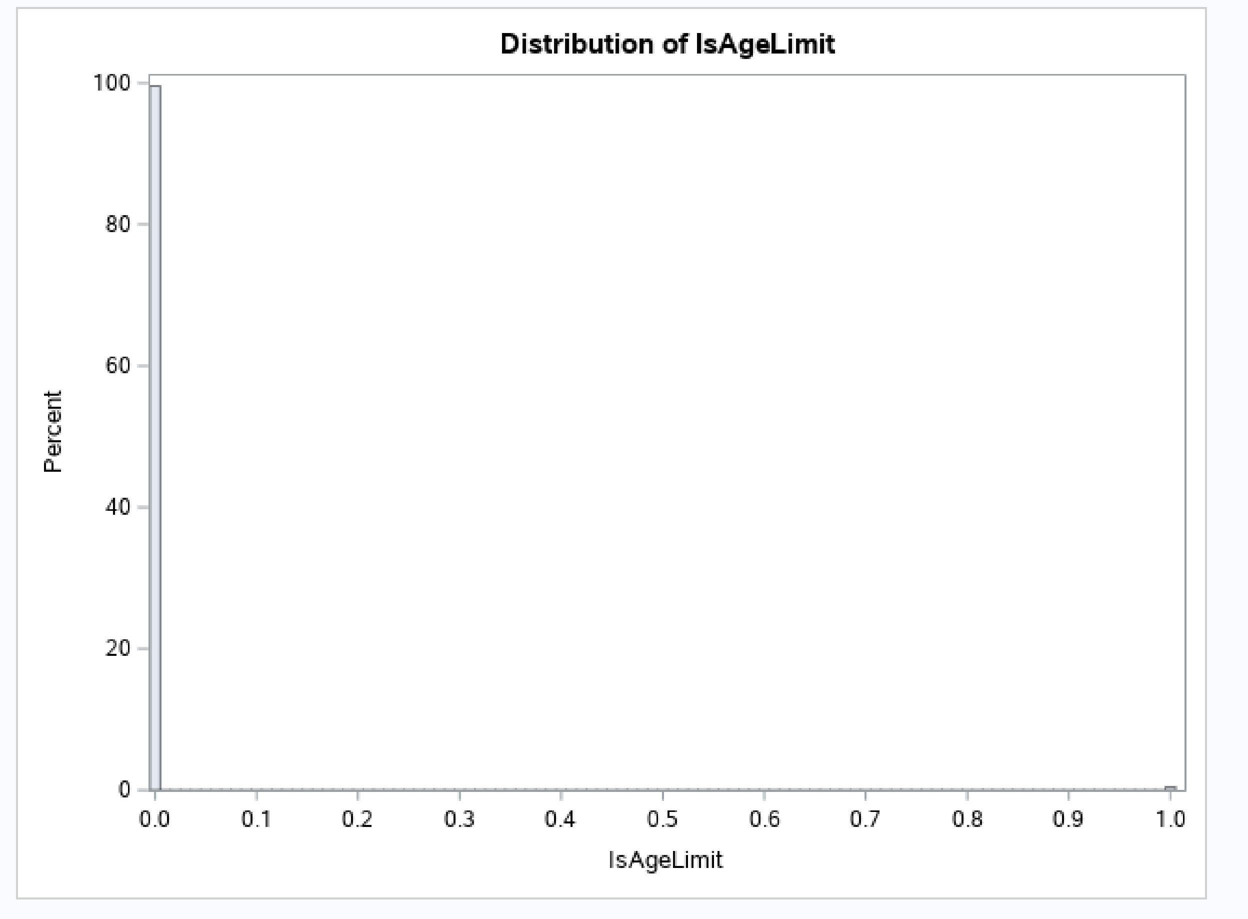

Distribution of IsAgeLimit
Percent
100
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
IsAgeLimit

Figure 20

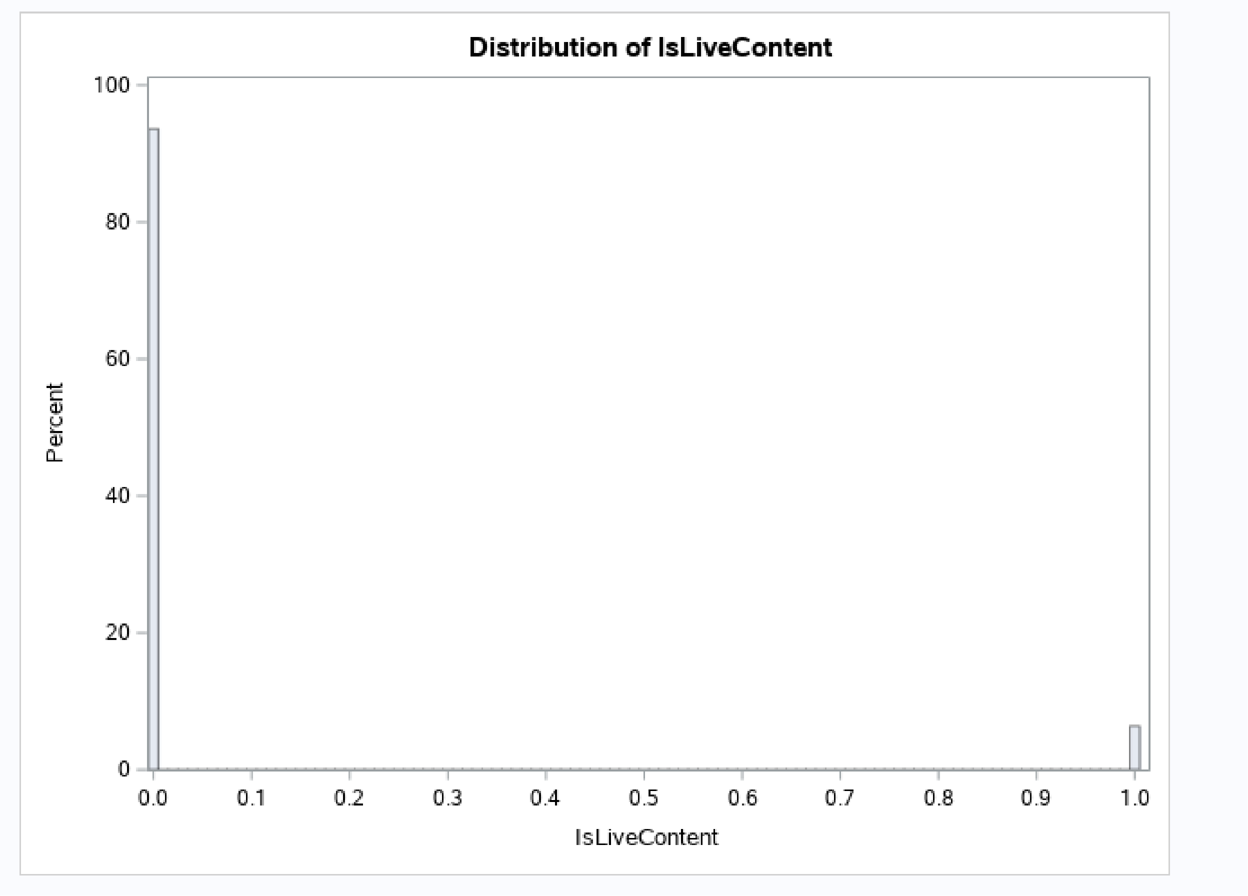

Distribution of IsLiveContent
100
80
60
40
20
0
Percent
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
IsLiveContent

Figure 21

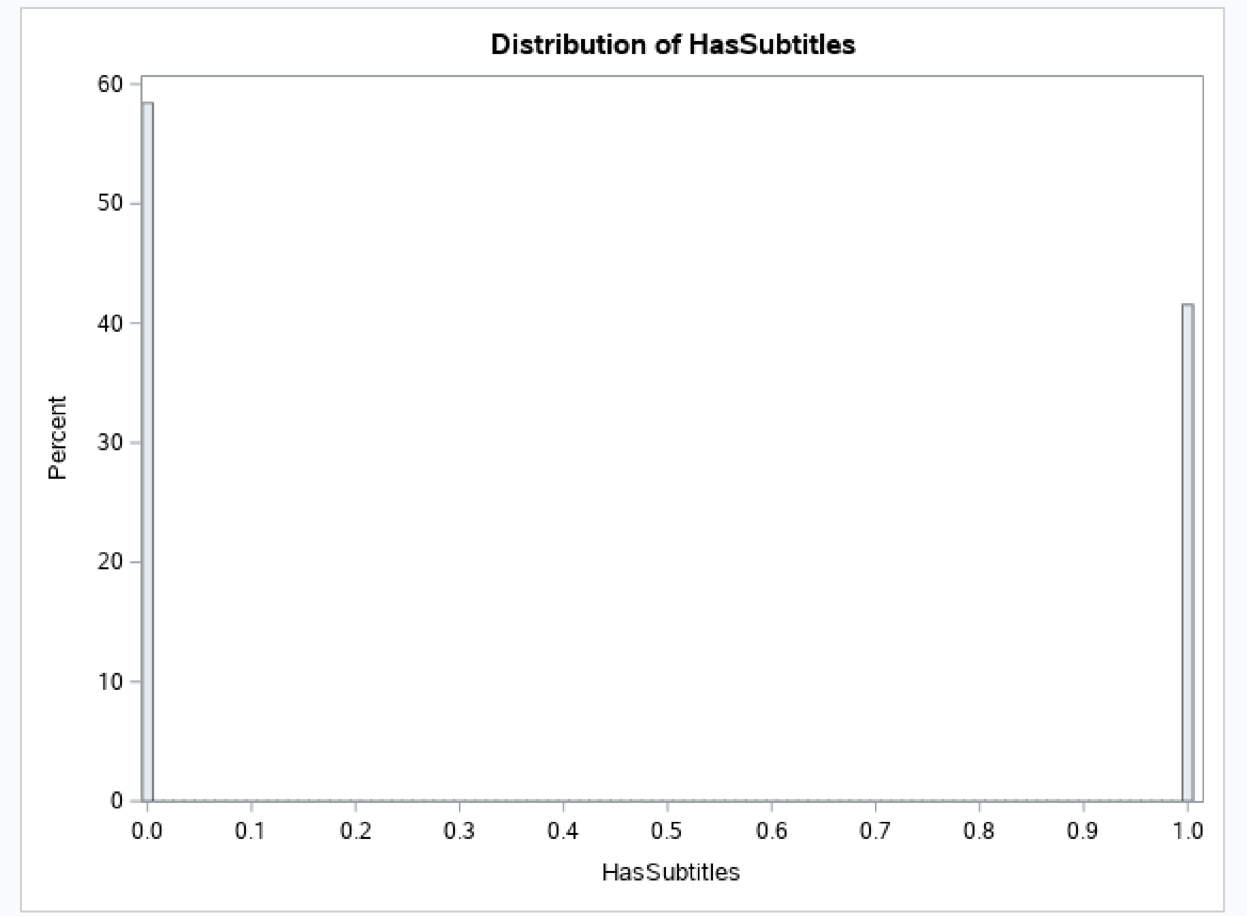

Distribution of HasSubtitles
Percent
60
50
40
30
20
10
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
HasSubtitles

Figure 22

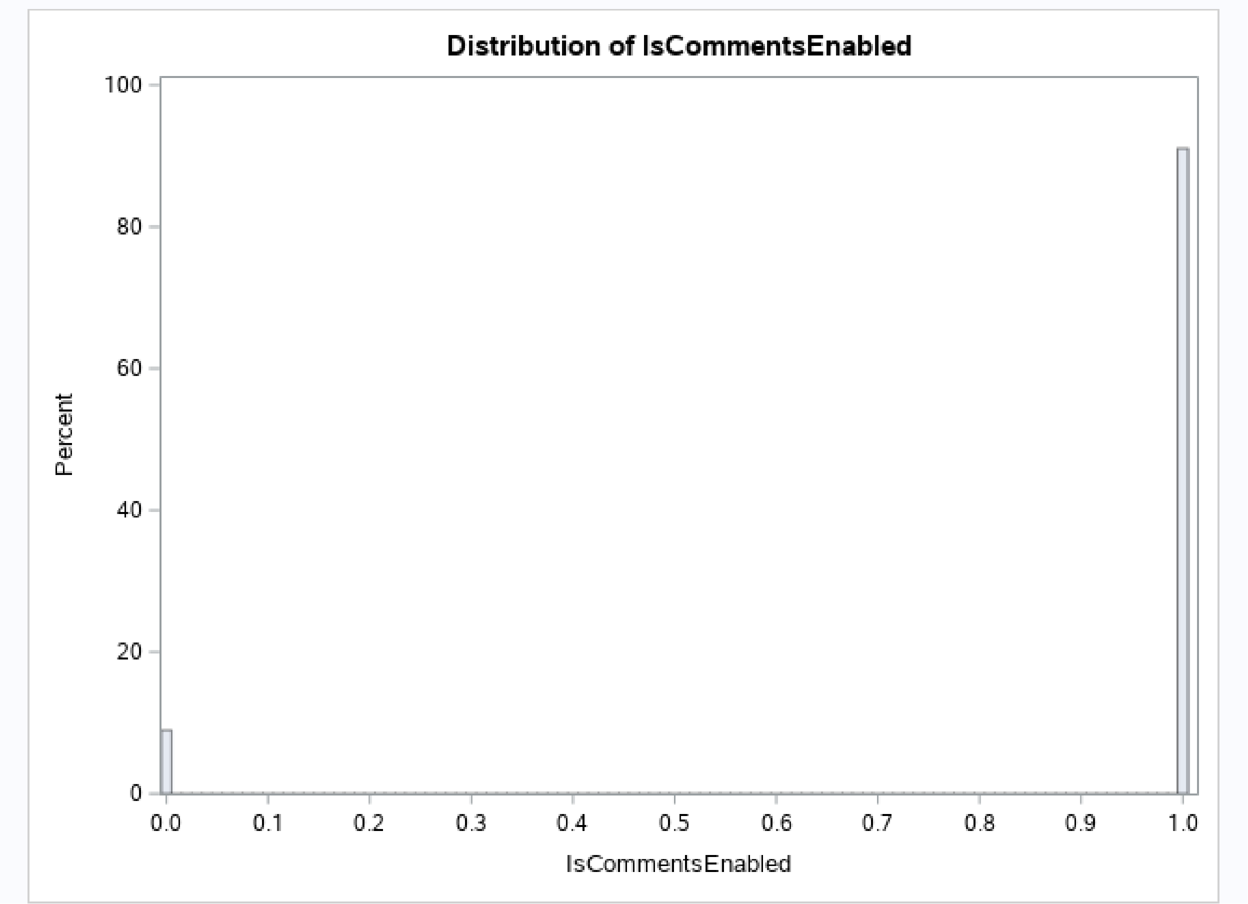

Distribution of IsCommentsEnabled
100
80
60
40
20
0
Percent
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
IsCommentsEnabled

Figure 23

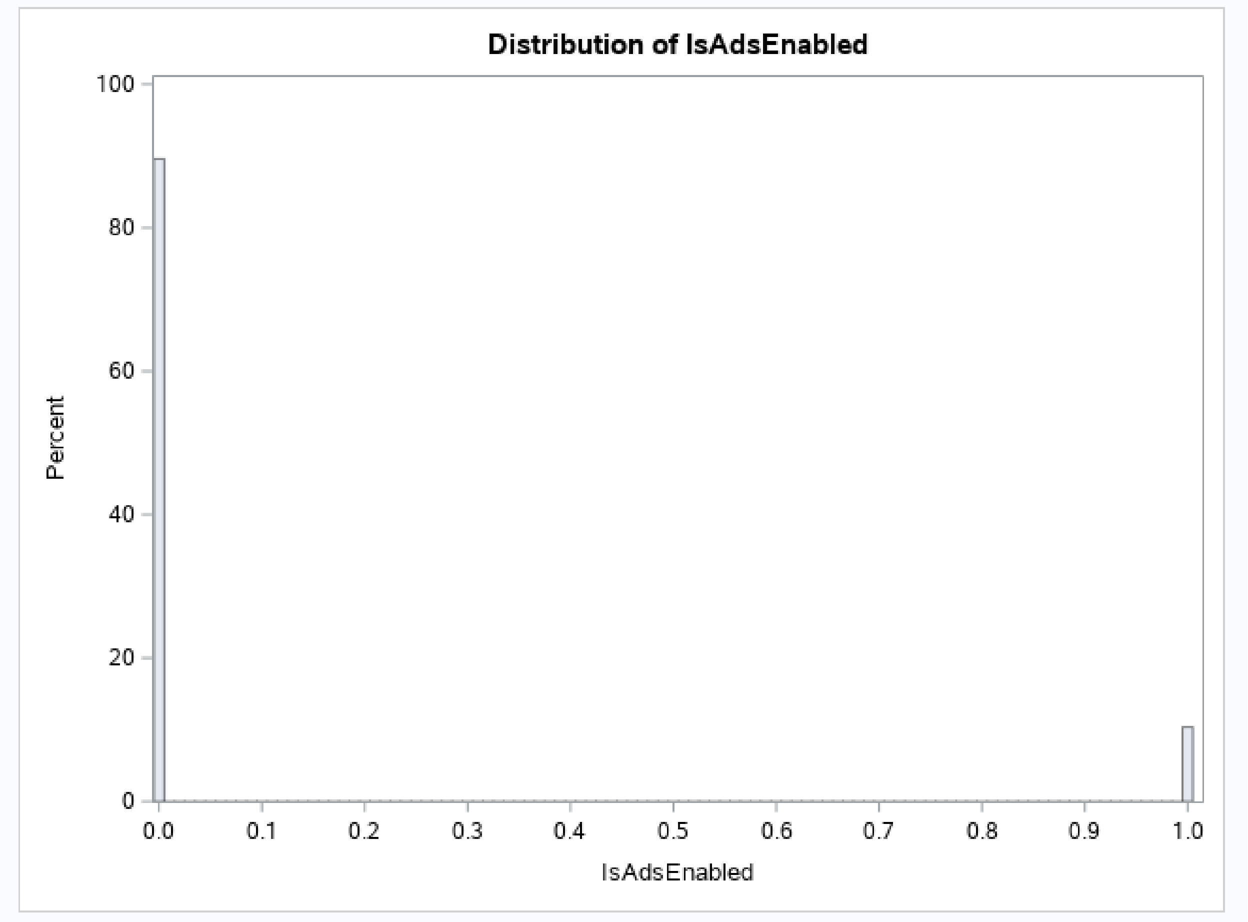
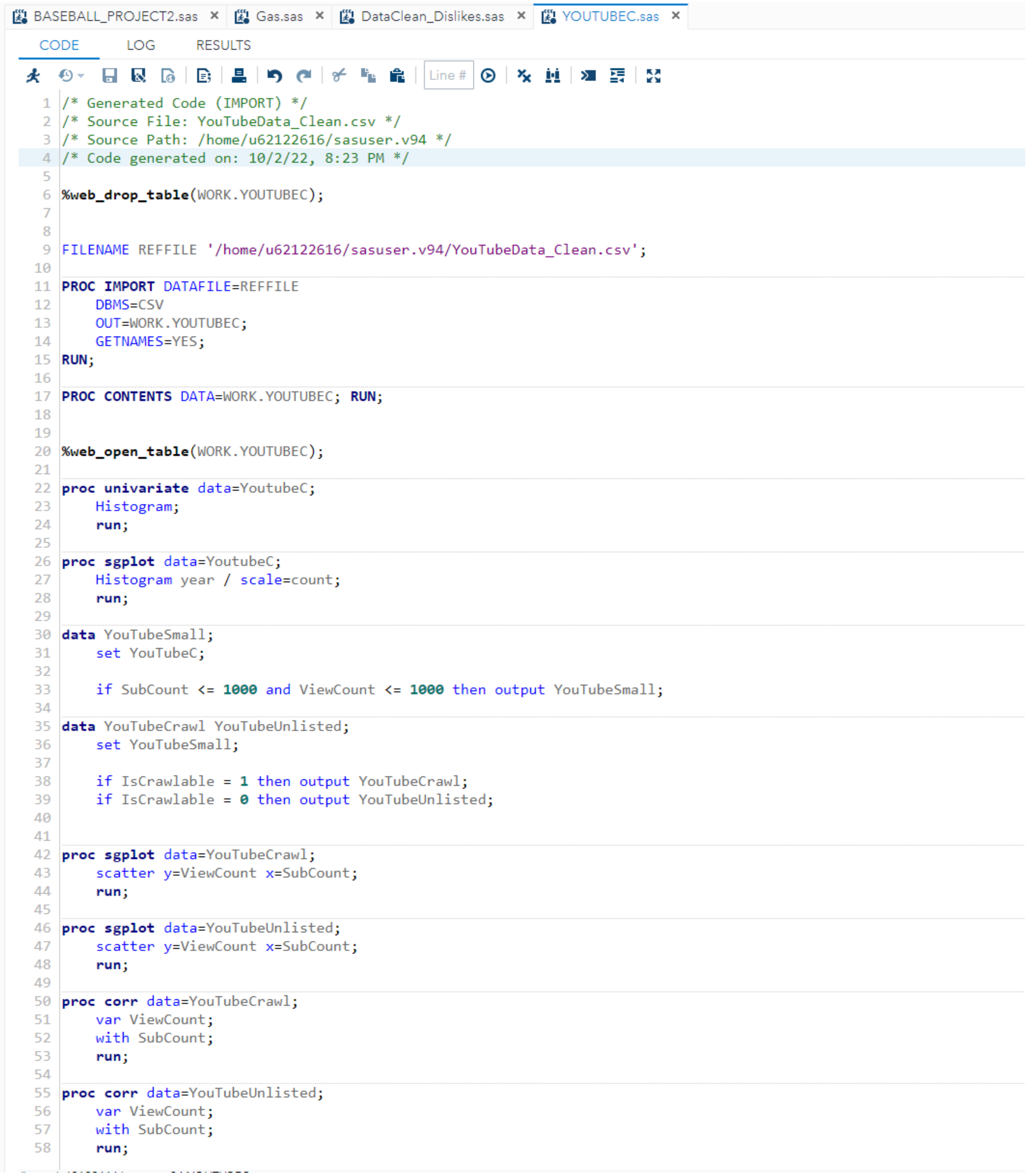

Distribution of IsAdsEnabled
Percent
100
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
IsAdsEnabled

Figure 24

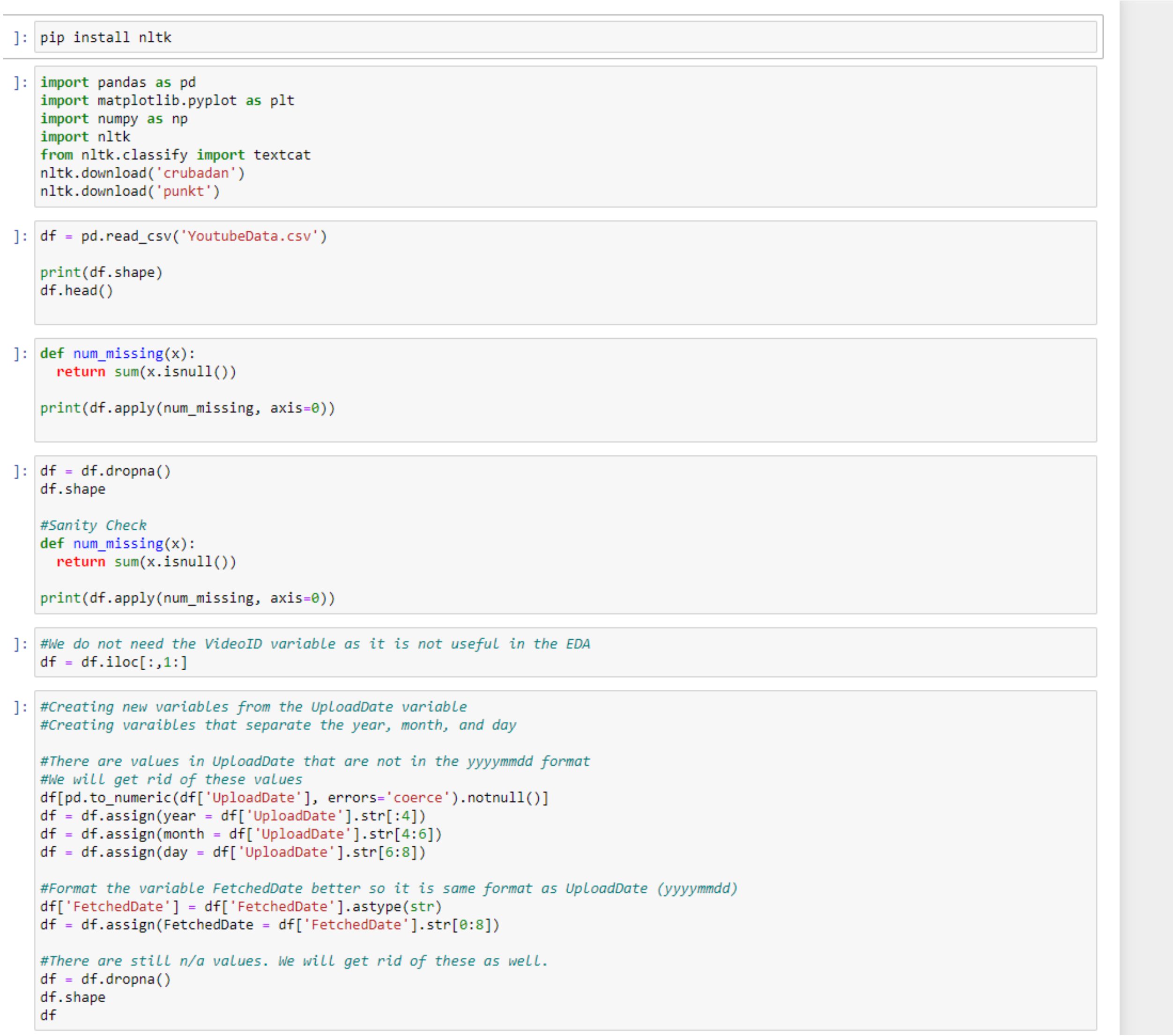
Code

Code for EDA



```
/* Generated Code (IMPORT) */
/* Source File: YouTubeData_Clean.csv */
/* Source Path: /home/u62122616/sasuser.v94 */
/* Code generated on: 10/2/22, 8:23 PM */

%web_drop_table(WORK.YOUTUBEC);


FILENAME REFFILE '/home/u62122616/sasuser.v94/YouTubeData_Clean.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=WORK.YOUTUBEC;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=WORK.YOUTUBEC; RUN;


%web_open_table(WORK.YOUTUBEC);

proc univariate data=YoutubeC;
    Histogram;
    run;

proc sgplot data=YoutubeC;
    Histogram year / scale=count;
    run;

data YouTubeSmall;
    set YouTubeC;

    if SubCount <= 1000 and ViewCount <= 1000 then output YouTubeSmall;

data YouTubeCrawl YouTubeUnlisted;
    set YouTubeSmall;

    if IsCrawlable = 1 then output YouTubeCrawl;
    if IsCrawlable = 0 then output YouTubeUnlisted;


proc sgplot data=YouTubeCrawl;
    scatter y=ViewCount x=SubCount;
    run;

proc sgplot data=YouTubeUnlisted;
    scatter y=ViewCount x=SubCount;
    run;

proc corr data=YouTubeCrawl;
    var ViewCount;
    with SubCount;
    run;

proc corr data=YouTubeUnlisted;
    var ViewCount;
    with SubCount;
    run;
```

Initial Data Cleaning



```
pip install nltk
```

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import nltk
from nltk.classify import textcat
nltk.download('crubadan')
nltk.download('punkt')
```

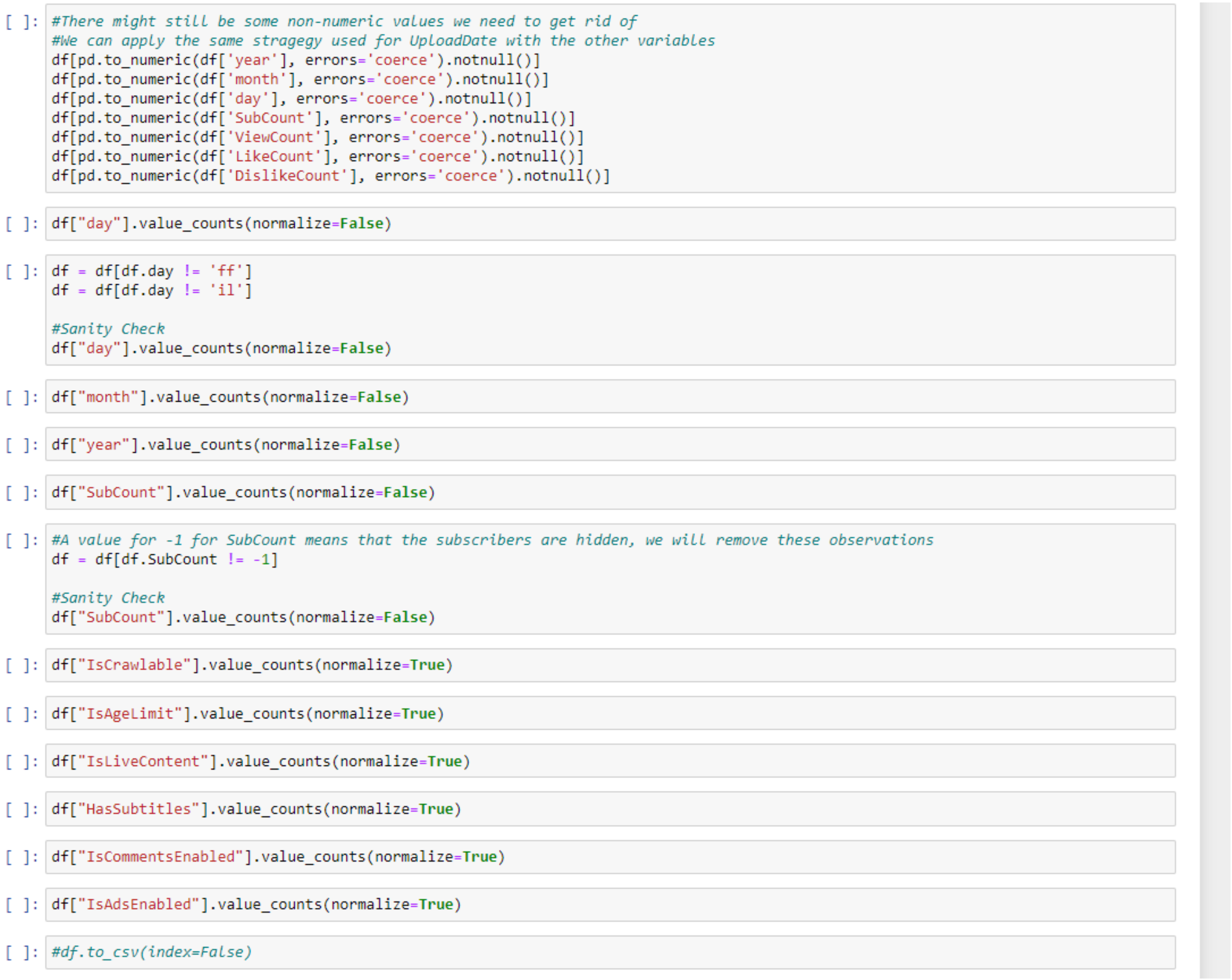
```
df = pd.read_csv('YoutubeData.csv')

print(df.shape)
df.head()
```

```
def num_missing(x):
  return sum(x.isnull())

print(df.apply(num_missing, axis=0))
```

```
df = df.dropna()
df.shape

#Sanity Check
def num_missing(x):
  return sum(x.isnull())

print(df.apply(num_missing, axis=0))
```

```
#We do not need the VideoID variable as it is not useful in the EDA
df = df.iloc[:,1:]
```

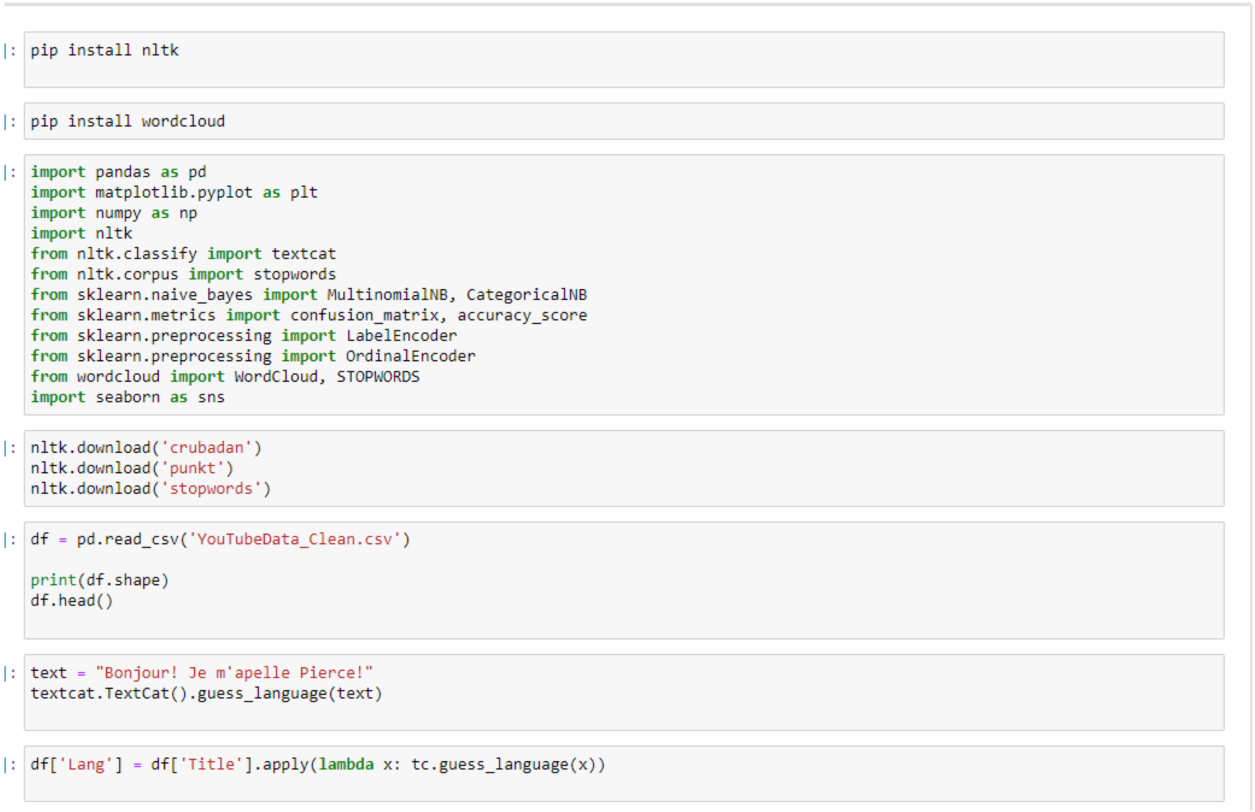
```
#Creating new variables from the UploadDate variable
#Creating varaibles that separate the year, month, and day

#There are values in UploadDate that are not in the yyyymmdd format
#We will get rid of these values
df[pd.to_numeric(df['UploadDate'], errors='coerce').notnull()]
df = df.assign(year = df['UploadDate'].str[:4])
df = df.assign(month = df['UploadDate'].str[4:6])
df = df.assign(day = df['UploadDate'].str[6:8])

#Format the variable FetchedDate better so it is same format as UploadDate (yyyymmdd)
df['FetchedDate'] = df['FetchedDate'].astype(str)
df = df.assign(FetchedDate = df['FetchedDate'].str[0:8])

#There are still n/a values. We will get rid of these as well.
df = df.dropna()
df.shape
df
```



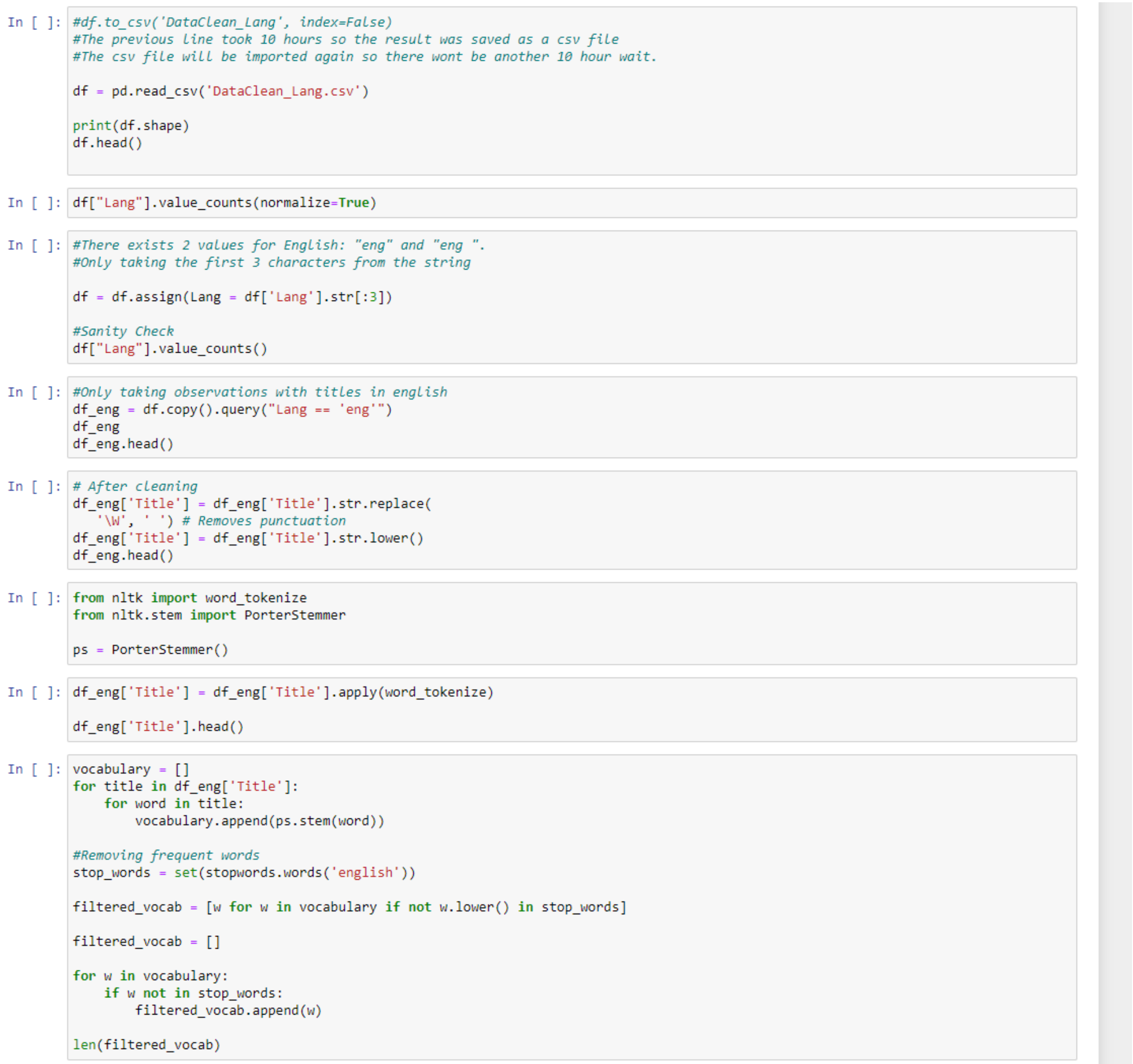
```
[ ]: #There might still be some non-numeric values we need to get rid of
     #We can apply the same stragegy used for UploadDate with the other variables
     df[pd.to_numeric(df['year'], errors='coerce').notnull()]
     df[pd.to_numeric(df['month'], errors='coerce').notnull()]
     df[pd.to_numeric(df['day'], errors='coerce').notnull()]
     df[pd.to_numeric(df['SubCount'], errors='coerce').notnull()]
     df[pd.to_numeric(df['ViewCount'], errors='coerce').notnull()]
     df[pd.to_numeric(df['LikeCount'], errors='coerce').notnull()]
     df[pd.to_numeric(df['DislikeCount'], errors='coerce').notnull()]

[ ]: df["day"].value_counts(normalize=False)

[ ]: df = df[df.day != 'ff']
     df = df[df.day != 'il']

     #Sanity Check
     df["day"].value_counts(normalize=False)

[ ]: df["month"].value_counts(normalize=False)

[ ]: df["year"].value_counts(normalize=False)

[ ]: df["SubCount"].value_counts(normalize=False)

[ ]: #A value for -1 for SubCount means that the subscribers are hidden, we will remove these observations
     df = df[df.SubCount != -1]

     #Sanity Check
     df["SubCount"].value_counts(normalize=False)

[ ]: df["IsCrawlable"].value_counts(normalize=True)

[ ]: df["IsAgeLimit"].value_counts(normalize=True)

[ ]: df["IsLiveContent"].value_counts(normalize=True)

[ ]: df["HasSubtitles"].value_counts(normalize=True)

[ ]: df["IsCommentsEnabled"].value_counts(normalize=True)

[ ]: df["IsAdsEnabled"].value_counts(normalize=True)

[ ]: #df.to_csv(index=False)
```

Code for the sentiment analysis/Naive Bayes Classification

(There are many parts where the data frame was converted to a csv as it sometimes took 10 hours to create that data frame. It was saved so that time would not have to be spent creating it again.)



```
]: pip install nltk

]: pip install wordcloud

]: import pandas as pd
   import matplotlib.pyplot as plt
   import numpy as np
   import nltk
   from nltk.classify import textcat
   from nltk.corpus import stopwords
   from sklearn.naive_bayes import MultinomialNB, CategoricalNB
   from sklearn.metrics import confusion_matrix, accuracy_score
   from sklearn.preprocessing import LabelEncoder
   from sklearn.preprocessing import OrdinalEncoder
   from wordcloud import WordCloud, STOPWORDS
   import seaborn as sns

]: nltk.download('crubadan')
   nltk.download('punkt')
   nltk.download('stopwords')

]: df = pd.read_csv('YouTubeData_Clean.csv')

   print(df.shape)
   df.head()

]: text = "Bonjour! Je m'apelle Pierce!"
   textcat.TextCat().guess_language(text)

]: df['Lang'] = df['Title'].apply(lambda x: tc.guess_language(x))
```



In [ ]:

```
#df.to_csv('DataClean_Lang', index=False)
#The previous line took 10 hours so the result was saved as a csv file
#The csv file will be imported again so there wont be another 10 hour wait.

df = pd.read_csv('DataClean_Lang.csv')

print(df.shape)
df.head()
```

In [ ]:

```
df["Lang"].value_counts(normalize=True)
```

In [ ]:

```
#There exists 2 values for English: "eng" and "eng ".
#Only taking the first 3 characters from the string

df = df.assign(Lang = df['Lang'].str[:3])

#Sanity Check
df["Lang"].value_counts()
```

In [ ]:

```
#Only taking observations with titles in english
df_eng = df.copy().query("Lang == 'eng'")
df_eng
df_eng.head()
```

In [ ]:

```
# After cleaning
df_eng['Title'] = df_eng['Title'].str.replace(
   '\W', ' ') # Removes punctuation
df_eng['Title'] = df_eng['Title'].str.lower()
df_eng.head()
```

In [ ]:

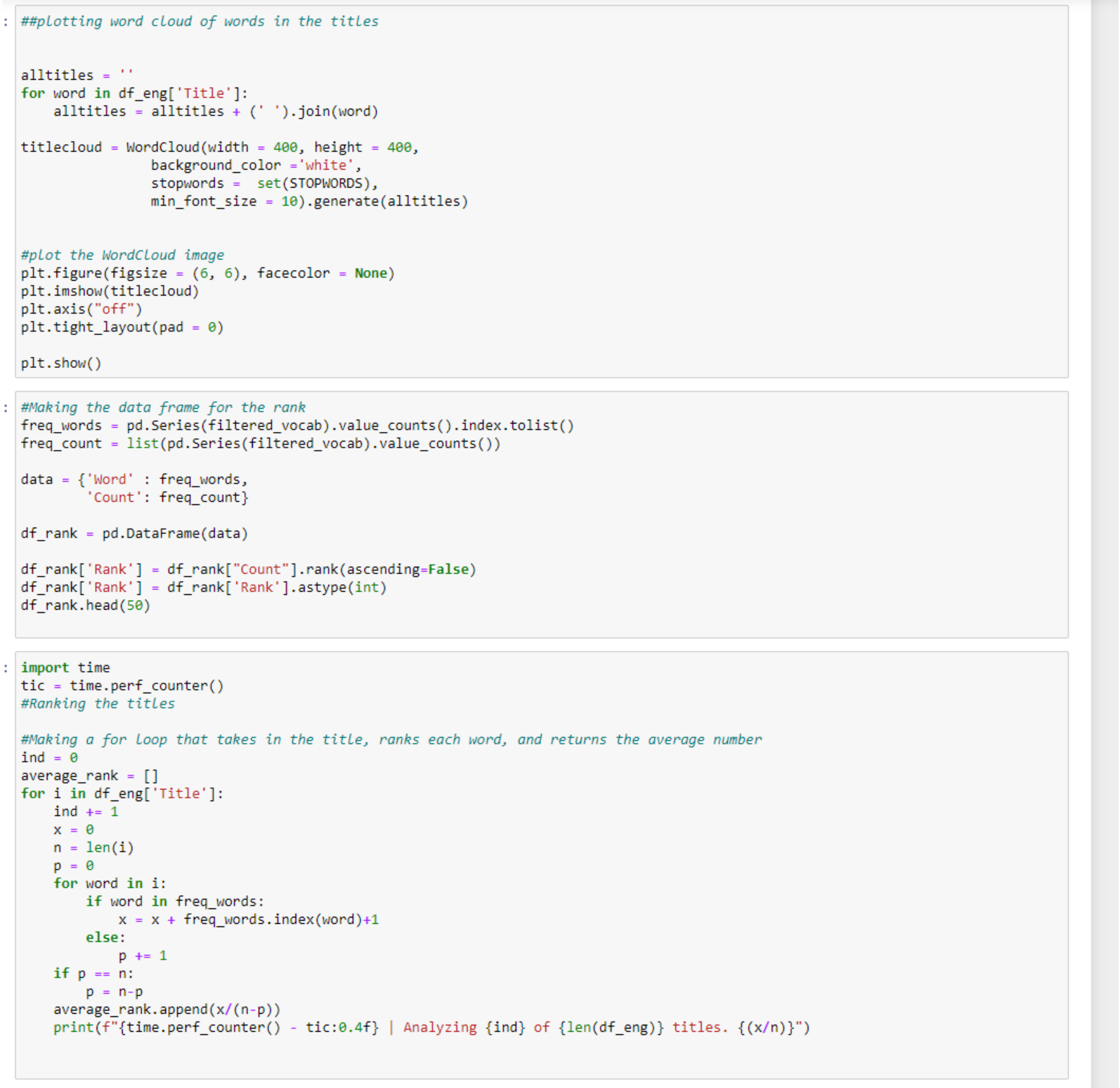
```
from nltk import word_tokenize
from nltk.stem import PorterStemmer

ps = PorterStemmer()
```

In [ ]:

```
df_eng['Title'] = df_eng['Title'].apply(word_tokenize)

df_eng['Title'].head()
```

In [ ]:

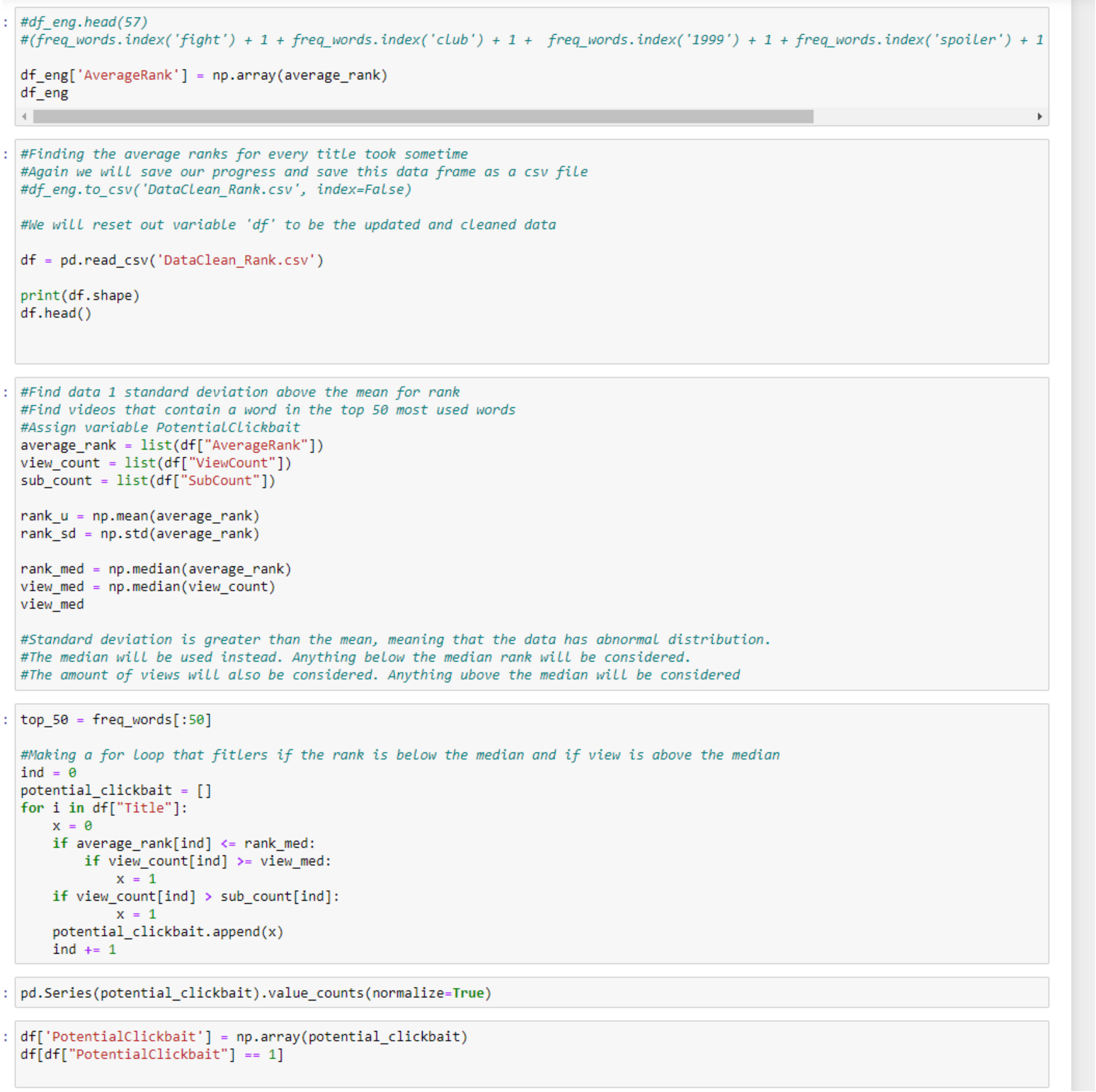
```
vocabulary = []
for title in df_eng['Title']:
    for word in title:
        vocabulary.append(ps.stem(word))

#Removing frequent words
stop_words = set(stopwords.words('english'))

filtered_vocab = [w for w in vocabulary if not w.lower() in stop_words]

filtered_vocab = []

for w in vocabulary:
    if w not in stop_words:
        filtered_vocab.append(w)

len(filtered_vocab)
```



```
##plotting word cloud of words in the titles


alltitles = ''
for word in df_eng['Title']:
    alltitles = alltitles + (' ').join(word)

titlecloud = WordCloud(width = 400, height = 400,
                background_color ='white',
                stopwords =  set(STOPWORDS),
                min_font_size = 10).generate(alltitles)


#plot the WordCloud image
plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(titlecloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

```
#Making the data frame for the rank
freq_words = pd.Series(filtered_vocab).value_counts().index.tolist()
freq_count = list(pd.Series(filtered_vocab).value_counts())

data = {'Word' : freq_words,
        'Count': freq_count}

df_rank = pd.DataFrame(data)

df_rank['Rank'] = df_rank["Count"].rank(ascending=False)
df_rank['Rank'] = df_rank['Rank'].astype(int)
df_rank.head(50)
```

```
import time
tic = time.perf_counter()
#Ranking the titles

#Making a for loop that takes in the title, ranks each word, and returns the average number
ind = 0
average_rank = []
for i in df_eng['Title']:
    ind += 1
    x = 0
    n = len(i)
    p = 0
    for word in i:
        if word in freq_words:
            x = x + freq_words.index(word)+1
        else:
            p += 1
    if p == n:
        p = n-p
    average_rank.append(x/(n-p))
    print(f"{time.perf_counter() - tic:0.4f} | Analyzing {ind} of {len(df_eng)} titles. {(x/n)}")
```



```
#df_eng.head(57)
#(freq_words.index('fight') + 1 + freq_words.index('club') + 1 +  freq_words.index('1999') + 1 + freq_words.index('spoiler') + 1

df_eng['AverageRank'] = np.array(average_rank)
df_eng
```

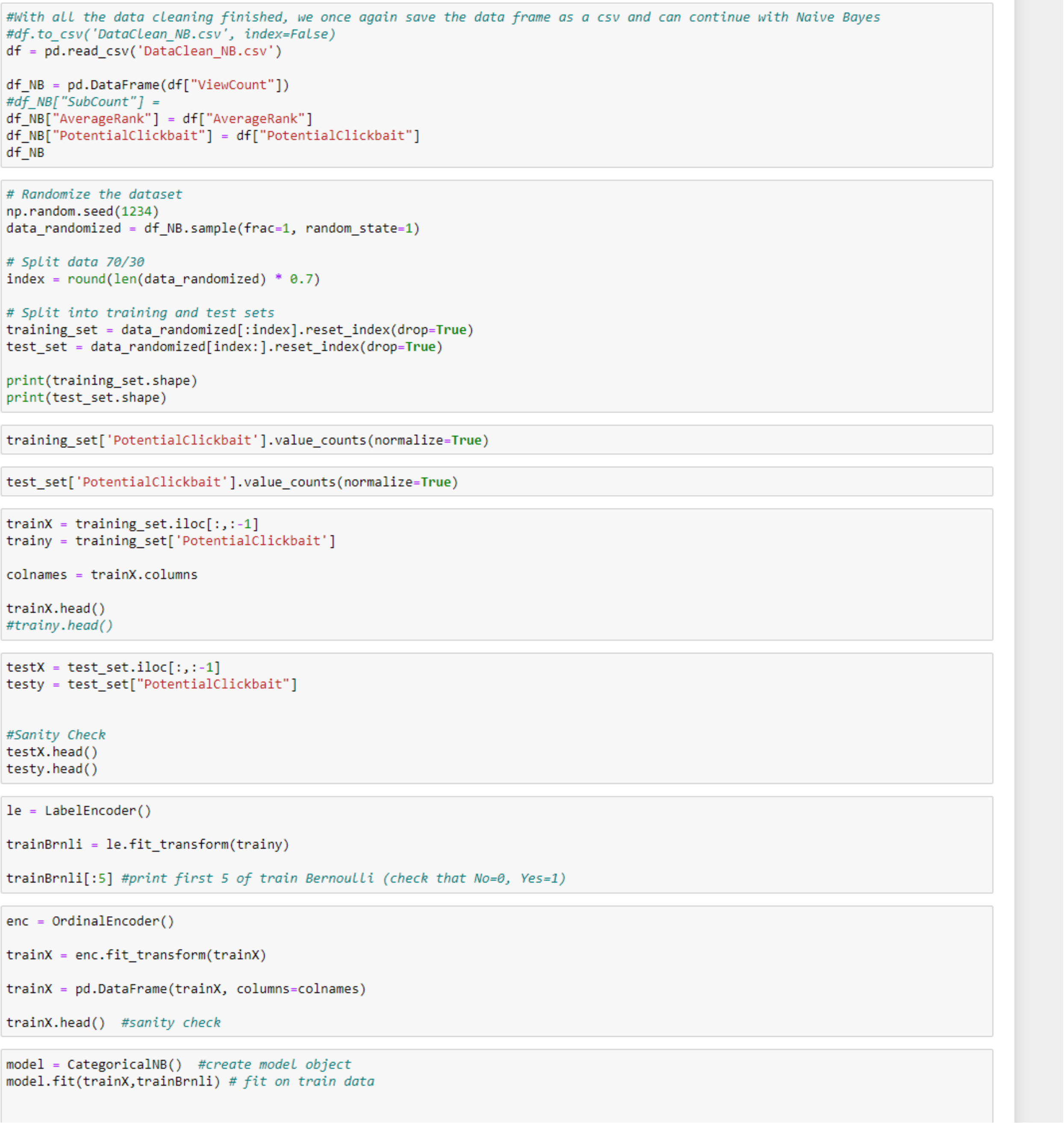
```
#Finding the average ranks for every title took sometime
#Again we will save our progress and save this data frame as a csv file
#df_eng.to_csv('DataClean_Rank.csv', index=False)

#We will reset out variable 'df' to be the updated and cleaned data

df = pd.read_csv('DataClean_Rank.csv')

print(df.shape)
df.head()
```

```
#Find data 1 standard deviation above the mean for rank
#Find videos that contain a word in the top 50 most used words
#Assign variable PotentialClickbait
average_rank = list(df["AverageRank"])
view_count = list(df["ViewCount"])
sub_count = list(df["SubCount"])

rank_u = np.mean(average_rank)
rank_sd = np.std(average_rank)

rank_med = np.median(average_rank)
view_med = np.median(view_count)
view_med

#Standard deviation is greater than the mean, meaning that the data has abnormal distribution.
#The median will be used instead. Anything below the median rank will be considered.
#The amount of views will also be considered. Anything above the median will be considered
```

```
top_50 = freq_words[:50]

#Making a for loop that fitlers if the rank is below the median and if view is above the median
ind = 0
potential_clickbait = []
for i in df["Title"]:
    x = 0
    if average_rank[ind] <= rank_med:
        if view_count[ind] >= view_med:
            x = 1
    if view_count[ind] > sub_count[ind]:
            x = 1
    potential_clickbait.append(x)
    ind += 1
```

```
pd.Series(potential_clickbait).value_counts(normalize=True)
```

```
df['PotentialClickbait'] = np.array(potential_clickbait)
df[df["PotentialClickbait"] == 1]
```



```
#With all the data cleaning finished, we once again save the data frame as a csv and can continue with Naive Bayes
#df.to_csv('DataClean_NB.csv', index=False)
df = pd.read_csv('DataClean_NB.csv')

df_NB = pd.DataFrame(df["ViewCount"])
#df_NB["SubCount"] =
df_NB["AverageRank"] = df["AverageRank"]
df_NB["PotentialClickbait"] = df["PotentialClickbait"]
df_NB
```

```
# Randomize the dataset
np.random.seed(1234)
data_randomized = df_NB.sample(frac=1, random_state=1)

# Split data 70/30
index = round(len(data_randomized) * 0.7)

# Split into training and test sets
training_set = data_randomized[:index].reset_index(drop=True)
test_set = data_randomized[index:].reset_index(drop=True)

print(training_set.shape)
print(test_set.shape)
```

```
training_set['PotentialClickbait'].value_counts(normalize=True)
```

```
test_set['PotentialClickbait'].value_counts(normalize=True)
```

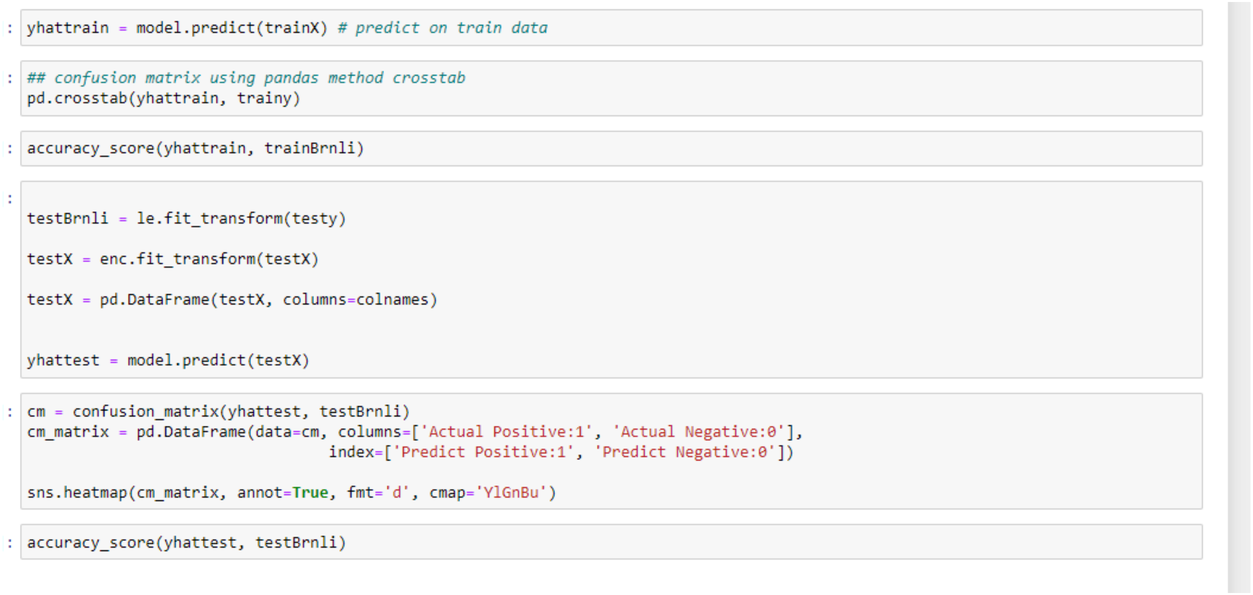
```
trainX = training_set.iloc[:,:-1]
trainy = training_set['PotentialClickbait']

colnames = trainX.columns

trainX.head()
#trainy.head()
```

```
testX = test_set.iloc[:,:-1]
testy = test_set["PotentialClickbait"]


#Sanity Check
testX.head()
testy.head()
```

```
le = LabelEncoder()

trainBrnli = le.fit_transform(trainy)

trainBrnli[:5] #print first 5 of train Bernoulli (check that No=0, Yes=1)
```

```
enc = OrdinalEncoder()

trainX = enc.fit_transform(trainX)

trainX = pd.DataFrame(trainX, columns=colnames)

trainX.head()  #sanity check
```

```
model = CategoricalNB()  #create model object
model.fit(trainX,trainBrnli) # fit on train data
```



```
yhattrain = model.predict(trainX) # predict on train data
```

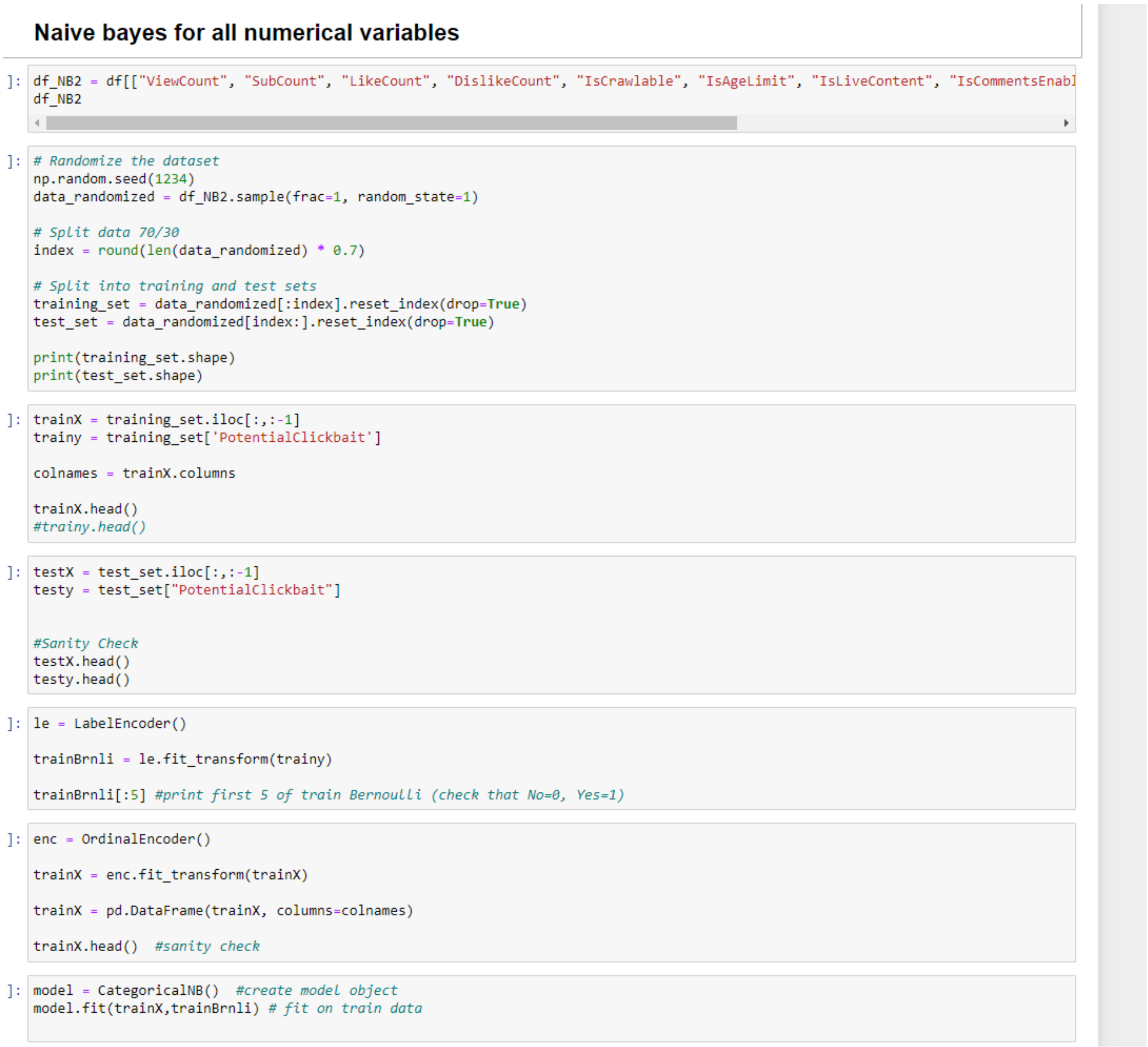
```
## confusion matrix using pandas method crosstab
pd.crosstab(yhattrain, trainy)
```

```
accuracy_score(yhattrain, trainBrnli)
```

```

testBrnli = le.fit_transform(testy)

testX = enc.fit_transform(testX)

testX = pd.DataFrame(testX, columns=colnames)


yhattest = model.predict(testX)
```

```
cm = confusion_matrix(yhattest, testBrnli)
cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual Negative:0'],
                                 index=['Predict Positive:1', 'Predict Negative:0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

```
accuracy_score(yhattest, testBrnli)
```



Naive bayes for all numerical variables

```
]: df_NB2 = df[["ViewCount", "SubCount", "LikeCount", "DislikeCount", "IsCrawlable", "IsAgeLimit", "IsLiveContent", "IsCommentsEnabl
   df_NB2
```

```
]: # Randomize the dataset
   np.random.seed(1234)
   data_randomized = df_NB2.sample(frac=1, random_state=1)

   # Split data 70/30
   index = round(len(data_randomized) * 0.7)

   # Split into training and test sets
   training_set = data_randomized[:index].reset_index(drop=True)
   test_set = data_randomized[index:].reset_index(drop=True)

   print(training_set.shape)
   print(test_set.shape)
```

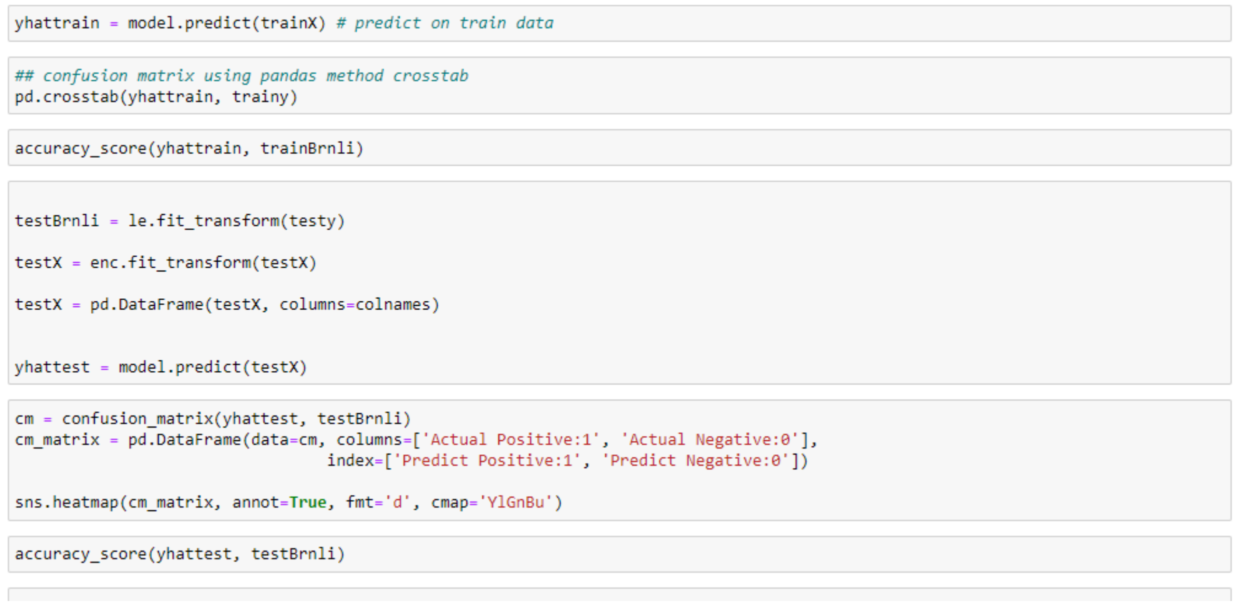
```
]: trainX = training_set.iloc[:,:-1]
   trainy = training_set['PotentialClickbait']

   colnames = trainX.columns

   trainX.head()
   #trainy.head()
```

```
]: testX = test_set.iloc[:,:-1]
   testy = test_set["PotentialClickbait"]


   #Sanity Check
   testX.head()
   testy.head()
```

```
]: le = LabelEncoder()

   trainBrnli = le.fit_transform(trainy)

   trainBrnli[:5] #print first 5 of train Bernoulli (check that No=0, Yes=1)
```

```
]: enc = OrdinalEncoder()

   trainX = enc.fit_transform(trainX)

   trainX = pd.DataFrame(trainX, columns=colnames)

   trainX.head()  #sanity check
```

```
]: model = CategoricalNB()  #create model object
   model.fit(trainX,trainBrnli) # fit on train data
```



```
yhattrain = model.predict(trainX) # predict on train data
```

```
## confusion matrix using pandas method crosstab
pd.crosstab(yhattrain, trainy)
```

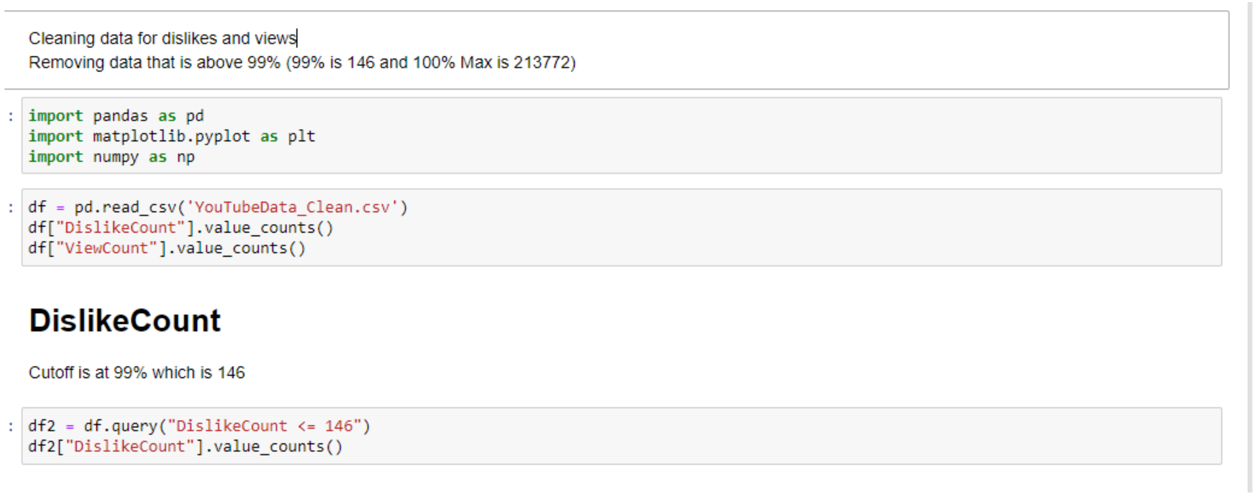
```
accuracy_score(yhattrain, trainBrnli)
```

```

testBrnli = le.fit_transform(testy)

testX = enc.fit_transform(testX)

testX = pd.DataFrame(testX, columns=colnames)


yhattest = model.predict(testX)
```

```
cm = confusion_matrix(yhattest, testBrnli)
cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual Negative:0'],
                                 index=['Predict Positive:1', 'Predict Negative:0'])

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

```
accuracy_score(yhattest, testBrnli)
```

Data Cleaning for RQ2



Cleaning data for dislikes and views
Removing data that is above 99% (99% is 146 and 100% Max is 213772)

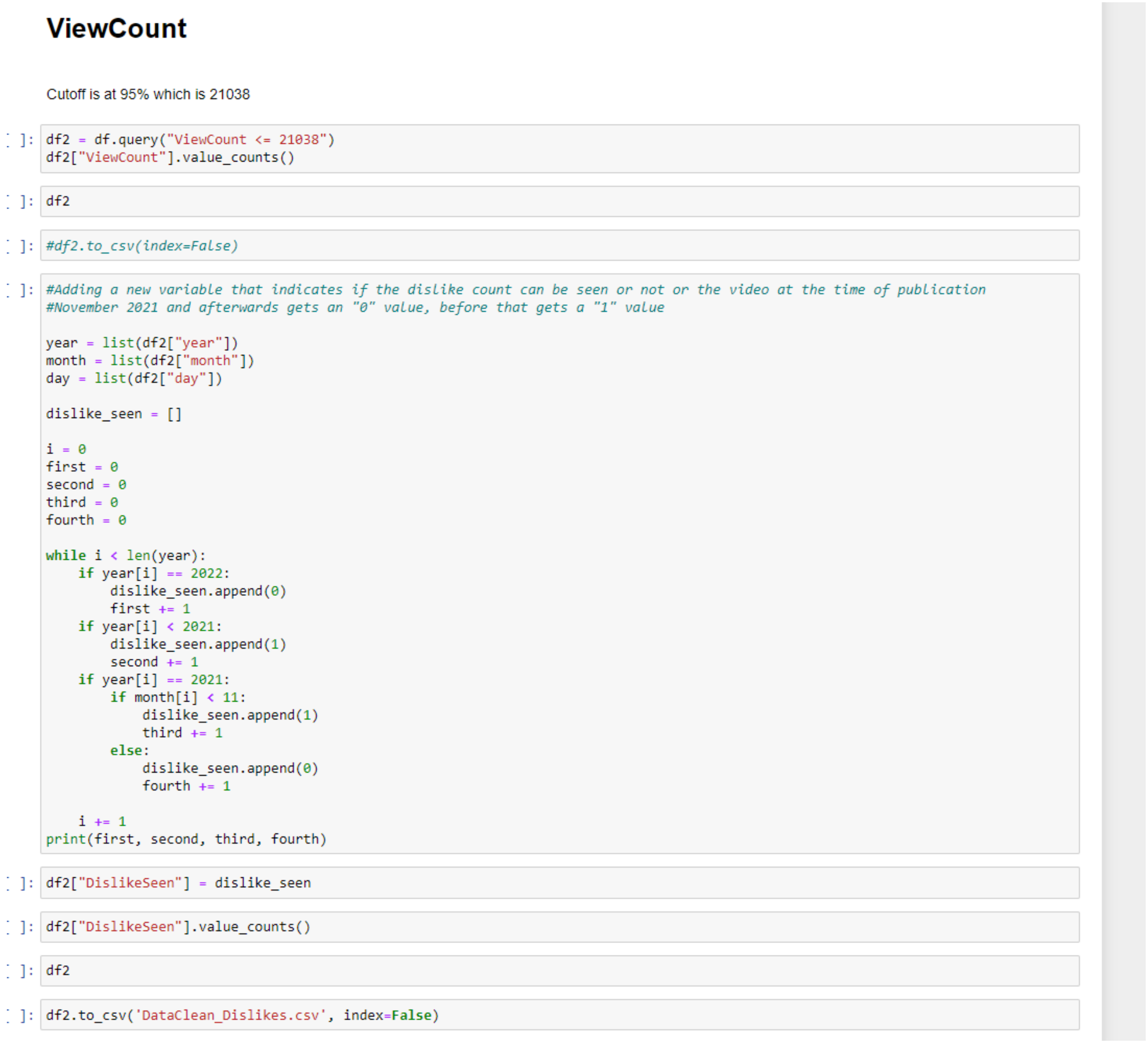
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
df = pd.read_csv('YouTubeData_Clean.csv')
df["DislikeCount"].value_counts()
df["ViewCount"].value_counts()
```

DislikeCount

Cutoff is at 99% which is 146

```
df2 = df.query("DislikeCount <= 146")
df2["DislikeCount"].value_counts()
```



ViewCount

Cutoff is at 95% which is 21038

```
df2 = df.query("ViewCount <= 21038")
df2["ViewCount"].value_counts()
```

```
df2
```

```
#df2.to_csv(index=False)
```

```
#Adding a new variable that indicates if the dislike count can be seen or not or the video at the time of publication
#November 2021 and afterwards gets an "0" value, before that gets a "1" value

year = list(df2["year"])
month = list(df2["month"])
day = list(df2["day"])

dislike_seen = []

i = 0
first = 0
second = 0
third = 0
fourth = 0

while i < len(year):
    if year[i] == 2022:
        dislike_seen.append(0)
        first += 1
    if year[i] < 2021:
        dislike_seen.append(1)
        second += 1
    if year[i] == 2021:
        if month[i] < 11:
            dislike_seen.append(1)
            third += 1
        else:
            dislike_seen.append(0)
            fourth += 1

    i += 1
print(first, second, third, fourth)
```

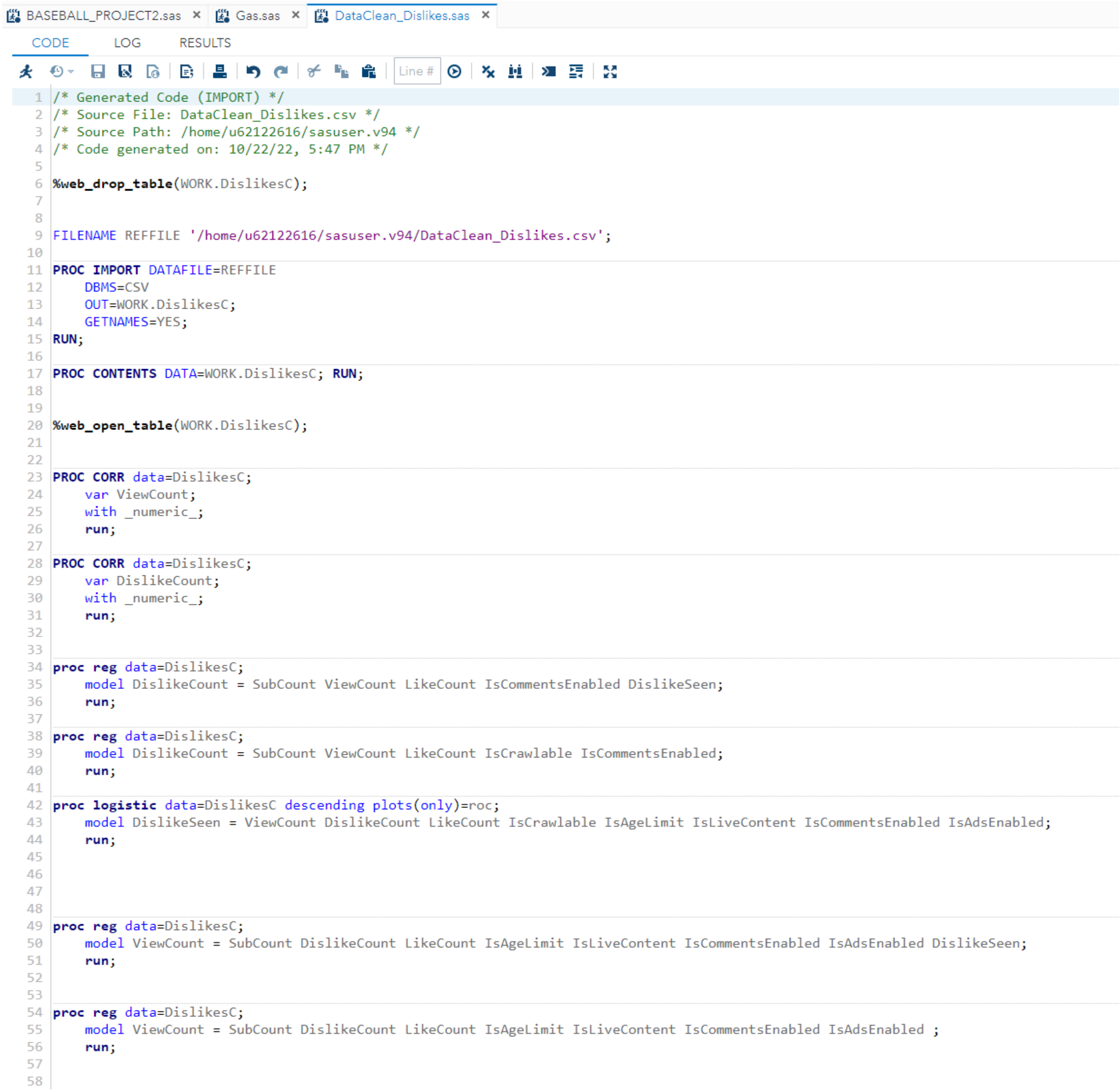
```
df2["DislikeSeen"] = dislike_seen
```

```
df2["DislikeSeen"].value_counts()
```

```
df2
```

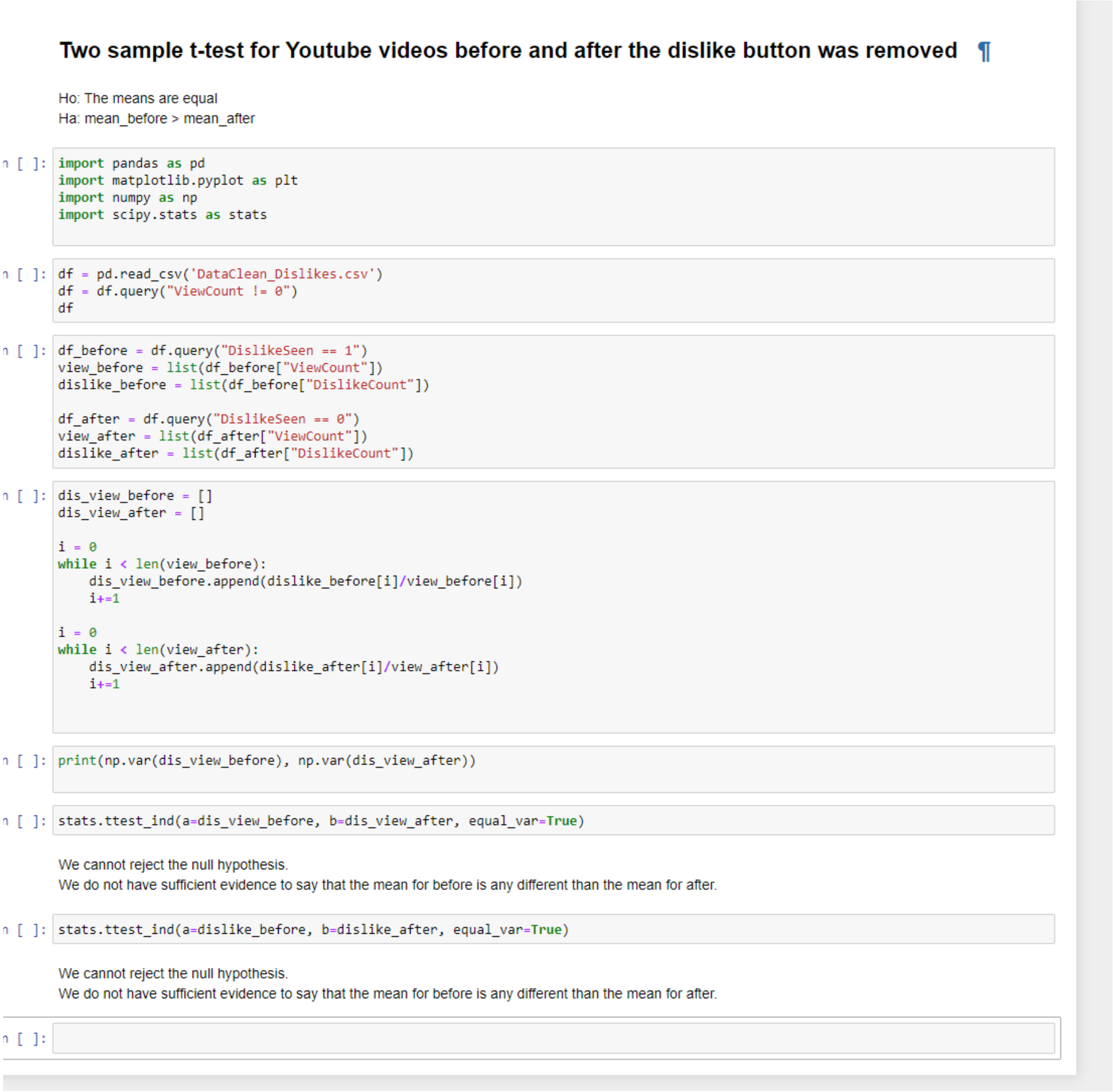
```
df2.to_csv('DataClean_Dislikes.csv', index=False)
```

Code for Linear/Logistic Regression



```
/* Generated Code (IMPORT) */
/* Source File: DataClean_Dislikes.csv */
/* Source Path: /home/u62122616/sasuser.v94 */
/* Code generated on: 10/22/22, 5:47 PM */

%web_drop_table(WORK.DislikesC);


FILENAME REFFILE '/home/u62122616/sasuser.v94/DataClean_Dislikes.csv';

PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=WORK.DislikesC;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=WORK.DislikesC; RUN;


%web_open_table(WORK.DislikesC);


PROC CORR data=DislikesC;
    var ViewCount;
    with _numeric_;
    run;

PROC CORR data=DislikesC;
    var DislikeCount;
    with _numeric_;
    run;


proc reg data=DislikesC;
    model DislikeCount = SubCount ViewCount LikeCount IsCommentsEnabled DislikeSeen;
    run;

proc reg data=DislikesC;
    model DislikeCount = SubCount ViewCount LikeCount IsCrawlable IsCommentsEnabled;
    run;

proc logistic data=DislikesC descending plots(only)=roc;
    model DislikeSeen = ViewCount DislikeCount LikeCount IsCrawlable IsAgeLimit IsLiveContent IsCommentsEnabled IsAdsEnabled;
    run;




proc reg data=DislikesC;
    model ViewCount = SubCount DislikeCount LikeCount IsAgeLimit IsLiveContent IsCommentsEnabled IsAdsEnabled DislikeSeen;
    run;


proc reg data=DislikesC;
    model ViewCount = SubCount DislikeCount LikeCount IsAgeLimit IsLiveContent IsCommentsEnabled IsAdsEnabled ;
    run;

```

Code for 2-sample t-test



Two sample t-test for Youtube videos before and after the dislike button was removed ¶

Ho: The means are equal
Ha: mean_before > mean_after

```
n [ ]: import pandas as pd
       import matplotlib.pyplot as plt
       import numpy as np
       import scipy.stats as stats
```

```
n [ ]: df = pd.read_csv('DataClean_Dislikes.csv')
       df = df.query("ViewCount != 0")
       df
```

```
n [ ]: df_before = df.query("DislikeSeen == 1")
       view_before = list(df_before["ViewCount"])
       dislike_before = list(df_before["DislikeCount"])

       df_after = df.query("DislikeSeen == 0")
       view_after = list(df_after["ViewCount"])
       dislike_after = list(df_after["DislikeCount"])
```

```
n [ ]: dis_view_before = []
       dis_view_after = []

       i = 0
       while i < len(view_before):
           dis_view_before.append(dislike_before[i]/view_before[i])
           i+=1

       i = 0
       while i < len(view_after):
           dis_view_after.append(dislike_after[i]/view_after[i])
           i+=1
```

```
n [ ]: print(np.var(dis_view_before), np.var(dis_view_after))
```

```
n [ ]: stats.ttest_ind(a=dis_view_before, b=dis_view_after, equal_var=True)
```

We cannot reject the null hypothesis.
We do not have sufficient evidence to say that the mean for before is any different than the mean for after.

```
n [ ]: stats.ttest_ind(a=dislike_before, b=dislike_after, equal_var=True)
```

We cannot reject the null hypothesis.
We do not have sufficient evidence to say that the mean for before is any different than the mean for after.

```
n [ ]:
```